Articulos en hábitat
Agosto 2024, Vol. 13, No. 1, pp. 1–45

Uso de redes neuronales en el pronóstico del PIB del subsector edificador en la ciudad de Bogotá

Neural networks in the GDP forecast of the building subsector in the city of Bogotá

Cristian Torres Casallas[a]
cristian.torres@habitatbogota.gov.co

Daniela Sedano Saenz[b]
daniela.sedano@habitatbogota.gov.co

Resumen

Este estudio evalúa el rendimiento de la metodología de redes neuronales en la estimación de pronósticos del Producto Interno Bruto (PIB) del subsector de la construcción de edificaciones en Bogotá, comparándola con enfoques comúnmente utilizados como ARIMA y VAR. Se emplea el promedio de las medidas de error cuadrático medio y error medio absoluto[1] para evaluar la precisión de los pronósticos. Los resultados indican que, dentro de la muestra analizada, los modelos ARIMA(3,0,1); ARIMA(2,0,4); ARIMA(3,0,2); VAR(2) con constante y sin parte determinística; VAR(1); VAR bayesiano(4); VAR bayesiano(3); Redes Neuronales Artificiales (RNA) con dos capas ocultas, tres rezagos, cinco neuronas por capa; RNA con dos capas ocultas, tres rezagos, ocho neuronas por capa; y una RNA con dos capas ocultas, tres rezagos y una neurona por capa son los de mejor desempeño. Estos hallazgos sugieren el potencial de las redes neuronales, lo cual podría tener implicaciones en el volumen de información disponible para la planificación económica de los actores del sector.

***Palabras clave*:** Redes neuronales, subsector edificador, construcción, ARIMA, VAR.

Abstract

This study evaluates the performance of neural network methodology in forecasting the Gross Domestic Product (GDP) of the building construction subsector in Bogotá, comparing it with commonly used approaches such as ARIMA and VAR. The mean squared error and mean absolute error[2] are used to evaluate forecast accuracy. The results indicate that, within the analyzed sample, the models ARIMA(3,0,1); ARIMA(2,0,4); ARIMA(3,0,2); VAR(2) with constant and without deterministic part; VAR(1); Bayesian VAR(4); Bayesian VAR(3); neural network with two hidden layers, three lags, five neurons per layer; neural network with two hidden layers, three lags, eight neurons per layer; and a neural network with two hidden layers, three lags, and one neuron per layer perform well. These findings suggest the potential of neural networks, which could have implications for the quality of information available for economic planning by sector stakeholders.

***Keywords*:** Neural network, construction sector, ARIMA, VAR.

1. Introducción

Las Redes Neuronales Artificiales (RNA) son modelos matemáticos diseñados para imitar el funcionamiento biológico de las neuronas. Su objetivo es emular procesos de aprendizaje y generalización,

[a] Ninguna
[b] Ninguna
[1] Estas dos medidas son estandarizadas con la función `scale` para poder ser combinadas mediante un promedio simple, lo que implica que se registran valores negativos
[2] These two measures are standardized using the `scale` function to be combined through a simple average, which implies that negative values are recorded

facilitando así la automatización de reglas para la toma de decisiones. Según investigaciones previas (Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe y White, 1989), se ha demostrado que las RNA pueden aproximar cualquier función continua con cualquier nivel de precisión. Esto tiene dos consecuencias clave: primero, no es necesario predecir la forma del proceso generador de datos; segundo, la eficacia de las RNA depende de la calidad y cantidad de datos de entrenamiento disponibles. En este caso, se considera que los datos disponibles no tienen aún la longitud suficiente para explotar al máximo el enfoque RNA. Sin embargo, se continúa con la investigación dada la importancia de evaluar el desempeño de este nuevo enfoque en un subsector que es relevante tanto económica como socialmente para la ciudad.

En términos sencillos, las Redes Neuronales Artificiales (RNA) son una familia de modelos utilizados para tareas de regresión, clasificación, pronóstico y reducción de dimensionalidad. Estas funcionan por la interconexión de elementos básicos llamados "neuronas", que imitan el proceso de comunicación neuronal denominado sinapsis[3]. A pesar de su simplicidad, las RNA tienen la capacidad de adaptarse, autoorganizarse y aprender, permitiéndoles manejar niveles significativos de complejidad.

En el contexto de series de tiempo, es posible modelar una variable mediante una RNA como un sistema con p nodos de entrada, h nodos en una capa oculta y un nodo de salida. Esta RNA toma como entrada p variables y las relaciona con una variable de salida, de esta manera tenemos que

$$y_t = \alpha_0 + \sum_{j=1}^{h} \alpha_j G\left(\beta_{0j} \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right) + \epsilon_t \tag{1}$$

Donde:

- y_t es el valor observado de la serie de tiempo en el instante t.
- y_{t-i} representa los valores rezagados de la serie de tiempo en el instante.
- α_j representa los pesos que van de la capa oculta a la capa de salida.
- β_{ij} representan los pesos que van de la capa de entrada a la capa oculta.
- G representa la función de activación o transferencia de la capa de entrada, la cual determina la salida de la capa oculta. Las funciones de activación que generalmente son usadas son la función logística para la capa de entrada y la función identidad para la capa de salida.
- h representa el número de neuronas en la capa oculta.
- p representa el número de neuronas en la capa de entrada.
- ϵ_t representa los errores aleatorios de modelo, los cuales se asumen que son independientes e idénticamente distribuidos con media cero y variancia constante.

La construcción en Bogotá ha sido históricamente un motor clave de la economía local, siendo fundamental en el desarrollo urbano y la generación de empleo de la ciudad. La actividad de la construcción representó en promedio (2021-2023) el 3,1 % del PIB y se divide en tres grandes subsectores: edificaciones residenciales y no residenciales, con un peso del 1,7 %; obras civiles, con un 0,9 %, y actividades especializadas de construcción, con 0,5 %. Por su parte, de acuerdo con las cifras de la Gran Encuesta Integrada de Hogares (GEIH), durante el 2023 actividad de la construcción participó con el 5,3 % de los ocupados del total de Bogotá y, específicamente, edificaciones participó con el 2,8 %. Las cifras de 2024 muestran que entre enero y marzo la participación promedio fue de 5,2 % (215.253 ocupados) para el sector de construcción y de 3,2 % en el subsector de edificaciones (131.123 ocupados). Lo anterior revela la importancia de esta actividad para el empleo del país.

De manera particular, el subsector de edificaciones desempeña un papel fundamental en la industria de la construcción en Bogotá contribuyendo en promedio al 58.2 % de su valor agregado anual, lo que equivale a una generación de alrededor de 6 billones de pesos. Este subsector se divide en dos categorías principales: residencial, que representa el 68.4 % de la actividad, y no residencial, que abarca el 31.6 %

[3]La sinapsis es la conexión especializada entre neuronas, ya sea entre dos neuronas de asociación, una neurona y una célula receptora, o entre una neurona y una célula efectora. En estos contactos se transmite el impulso nervioso, iniciándose con una descarga química que genera una corriente eléctrica en la membrana de la célula emisora (presináptica). Una vez que este impulso alcanza el extremo del axón, la conexión excita o inhibe la acción de otra célula receptora (postsináptica).

restante. Para calcular el Producto Interno Bruto (PIB) en Colombia, el Departamento Administrativo Nacional de Estadística (DANE) emplea el método de causación rezagada. Este enfoque sugiere que el valor agregado de cada proyecto en cada período se tome en la etapa en la que se encuentra durante la medición.

Tabla 1: Proceso de causación, coeficientes de incidencia, por destino, según capitulo constructivo

Capitulo constructivo	Grupo 1[a]	Grupo 2[b]	Grupo 3[c]
Preliminares[d]	10	13	26
Estructura[e]	20	27	36
Mampostería	15	10	8
Acabados 1[f]	45	43	24
Acabados 2[g]	9	6	5
Acabados 3[h]	1	1	1

[a] Vivienda unifamiliar y multifamiliar.
[b] Oficinas, locales, centros comerciales, centros de salud, hospitales, sedes institucionales y similares.
[c] Instalaciones industriales y bodegas.
[d] Incluye el proceso excavación y cimentación.
[e] Incluye la construcción de la cubierta.
[f] Carpintería, metálica y de madera, pisos, enchapes, recubrimientos de muros y cielo raso.
[g] Pintura, instalación de equipos y alfombras, vidrios y espejos, instalación de apliques.
[h] Remates, aseo y limpieza.

Las relaciones temporales, generalmente utilizadas para modelar el subsector de la construcción, se fundamentan en el siguiente esquema teórico de evolución de un proyecto constructivo. A partir de este esquema, se identifican como posibles predictores rezagados de la actividad: las ventas, el licenciamiento, el área en proceso y las iniciaciones.

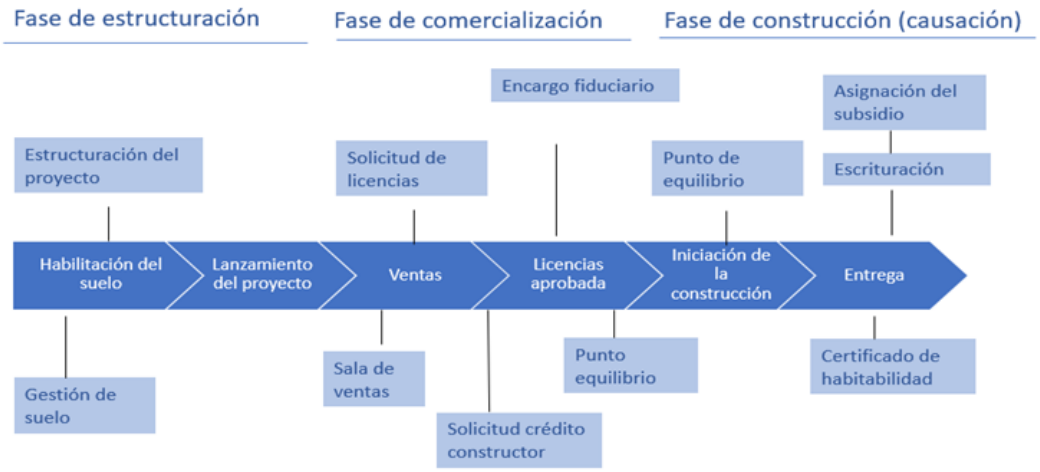

Figura 1: Fases del proceso constructivo del sector de edificaciones

Entre las aplicaciones realizadas en Colombia, destaca la llevada a cabo por Sáenz en 2009 sobre el PIB del país. Esta investigación revela relaciones no lineales en el proceso generador de datos, las cuales son capturadas de manera exitosa por una Red Neuronal Artificial (RNA). Sin embargo, bajo este enfoque, no es posible identificar la fuente de estas relaciones no lineales. Los resultados muestran que un modelo ampliado que incluye la tasa de interés de los Certificados de Depósito a 90 días mejora los pronósticos en al menos dos períodos, lo que indica que esta variable es relevante para el comportamiento del PIB. Además, se demuestra la eficacia de la metodología Rolling [4] al evaluar el desempeño de los modelos en términos de pronósticos. No obstante, esfuerzos como los de Iffat A. Gheyas en 2009 proporcionan evidencia de que esta superioridad solo es significativa a corto plazo.

[4]La metodología rolling (o de "ventana deslizante") es una técnica utilizada en el análisis de series de tiempo que permite calcular estadísticas o aplicar funciones sobre un subconjunto de datos dentro de una ventana que se desplaza a lo largo del tiempo. Esta técnica es útil para analizar cómo cambian las propiedades de la serie de tiempo a medida que avanzamos en el tiempo.

2. Desarrollo

En esta sección, el objetivo es mostrar el proceso de selección de las especificaciones. Esta selección se basa en la evaluación de dos métricas clave: el error cuadrático medio y el error medio absoluto. Para ello, se lleva a cabo el proceso detallado en la Tabla 2, utilizando un modelo $VAR(1)$. Este proceso implica generar una primera estimación con el conjunto completo de datos hasta diciembre de 2022, una segunda con los datos desde el segundo dato más antiguo hasta el cuarto trimestre de 2022, y así sucesivamente, hasta eliminar las cinco observaciones más antiguas. De esta manera, es posible evaluar el desempeño del modelo bajo varios conjuntos de datos.

Tabla 2: Ejemplo cálculo pronóstico modelo VAR(1) de forma iterativa

Matriz de datos	Especificación	Pronóstico
$\begin{pmatrix} x_{11} & \dots & x_{p1} \\ \vdots & \dots & \vdots \\ x_{1n} & \dots & x_{pn} \end{pmatrix}$	$X_t = A_0 + A_1 Y_{t-1} + \epsilon_t$	$(X_{1,t+1}, X_{1,t+2} \dots, X_{1,t+10})$
$\begin{pmatrix} x_{12} & \dots & x_{p2} \\ \vdots & \dots & \vdots \\ x_{1n} & \dots & x_{pn} \end{pmatrix}$	$X_t = A_0 + A_1 Y_{t-1} + \epsilon_t$	$(X_{2,t+1}, X_{2,t+2} \dots, X_{2,t+10})$
$\vdots$	$\vdots$	$\vdots$
$\begin{pmatrix} x_{1k} & \dots & x_{pk} \\ \vdots & \dots & \vdots \\ x_{1n} & \dots & x_{pn} \end{pmatrix}$	$X_t = A_0 + A_1 Y_{t-1} + \epsilon_t$	$(X_{k,t+1}, X_{k,t+2} \dots, X_{k,t+10})$
Promedio		$(\bar{X}_{t+1}, \bar{X}_{k,t+2} \dots, \bar{X}_{k,t+10})$

A continuación, se presentan los principales resultados de las estimaciones por metodología. Es importante tener en cuenta que estos resultados son temporales debido a posibles cambios en los datos asociados a actualizaciones metodológicas realizadas por el DANE. Además, mejoras en las técnicas, tanto en estimación como en comparación, pueden influir en los resultados.

2.1. Enfoque ARIMA

Siguiendo a (Hyndman & Athanasopoulos 2018) y (Wilcox 2012), antes de iniciar con este enfoque, es necesario eliminar los datos atípicos. Para identificarlos, se utiliza un análisis gráfico básico y la prueba de Rosner para series de tiempo, implementada mediante la función `rosnerTest` del paquete `EnvStats` (EnvStats Development Team 2023). Este ejercicio indica que existen cinco trimestres candidatos a configurarse como atípico. Sin embargo, solo hay evidencia estadística que permite catalogar al segundo trimestre de 2020 como un valor atípico. Como medida correctiva, se sustituye este punto por la mediana de ese trimestre.

Figura 2: Evolución temporal del PIB de edificaciones en Bogotá

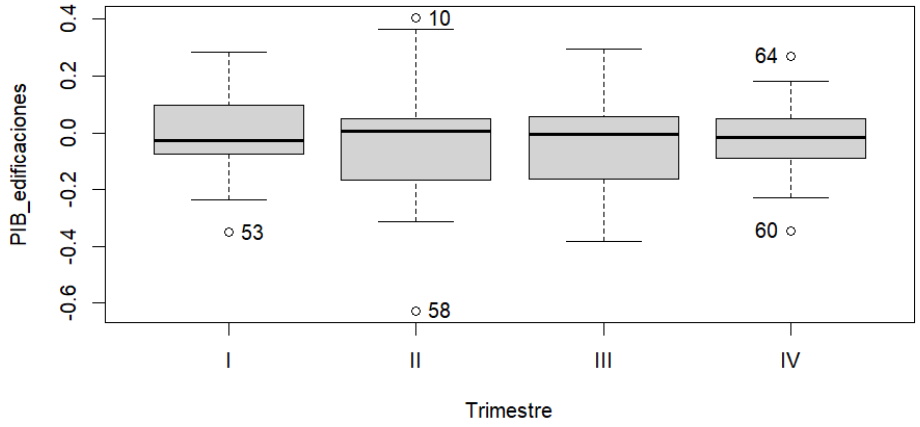

Figura 3: Diagrama de caja del PIB de edificaciones por trimestre

Para realizar las estimaciones se utiliza el enfoque Box-Jenkins (Box & Jenkins 1970), que ha sido fundamental para modelar series temporales, enfocándose en la identificación, estimación y verificación de modelos ARIMA. Un modelo $ARIMA(p, d, q)$ para el PIB se puede expresar como:

$$(1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p)(1 - L)^d PIB_t = (1 + \theta_1 L + \theta_2 L^2 + \ldots + \theta_q L^q) Z_t \tag{2}$$

Donde Z_t busca capturar patrones y efectos residuales en los datos de series temporales que no se pueden explicar mediante autorregresión o diferenciación. De esta manera, el valor estimado del PIB en el momento t mediante un modelo ARIMA se estima como una combinación lineal de sus valores pasados, otorgando mayor peso a los valores más recientes.

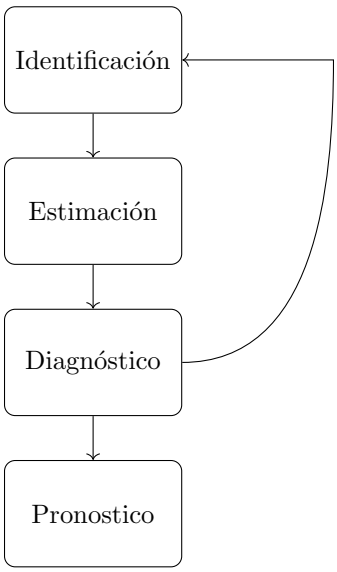

Figura 4: Esquema de la metodología Box-Jenkins.

La Figura 4 muestra el esquema básico de la metodología Box-Jenkins, la cual es aplicable únicamente a series estacionarias e incluye tres etapas iterativas. Si el modelo estimado no cumple con tres requisitos básicos: errores normales, no correlación serial y estabilidad, la especificación es desechada y se vuelve al paso de la identificación. Para determinar la estacionariedad de las series, se utiliza la función `ur.df` del paquete `urca` (R Core Team 2023). En cuanto a las pruebas de diagnóstico, se emplean las funciones `ur.df`, `ur.df` y `ur.df` del paquete `urca`.

Para la diferencia anual del logaritmo del PIB de edificaciones, la prueba de Dickey-Fuller indica, con un nivel de significación del cinco por ciento, que existe evidencia estadística para rechazar la hipótesis nula de raíz unitaria [5] con lo cual se considera adecuado el uso de la metodología Box-Jenkins.

En la identificación del modelo, se emplea el análisis gráfico de las funciones de autocorrelación simple y parcial, junto con el comando `auto.arima` del paquete `forecast`, el cual selecciona la mejor especificación de modelo ARIMA basada en el criterio de información bayesiano (BIC). La Figura 5 muestra una correlación significativa en el tercer rezago de la función de autocorrelación simple, sugiriendo la inclusión de un componente AR de tres períodos. Además, la función de autocorrelación parcial indica una correlación significativa en el cuarto rezago, después de eliminar la influencia de los rezagos intermedios, lo que podría indicar la presencia de un componente de media móvil (MA) en el modelo. Basándonos en estos resultados, se propone explorar el desempeño de modelos ARIMA que combinan tres rezagos autoregresivos y cuatro de media móvil y sus intermedios. Según `auto.arima`, el modelo más adecuado especifica un componente autoregresivo de orden 1, un componente de media móvil de orden 1, un término de media móvil y uno autoregresivo que corrige los errores de predicción cada 4 períodos, es decir, ARIMA(1,0,1)(1,0,1)[4].

[5] Se estimaron nueve especificaciones de la prueba bajo los esquemas de tendencia, tendencia estocástica y sin ningún tipo de tendencia, combinados con modelos que incluyen entre cero y dos rezagos. De estos resultados la prueba de raíz unitaria solo es aceptada al cinco por ciento bajo la especificación de tendencia y dos rezagos, por lo cual se considera que existe evidencia para considerar adecuado el uso de la metodología Box-Jenkins

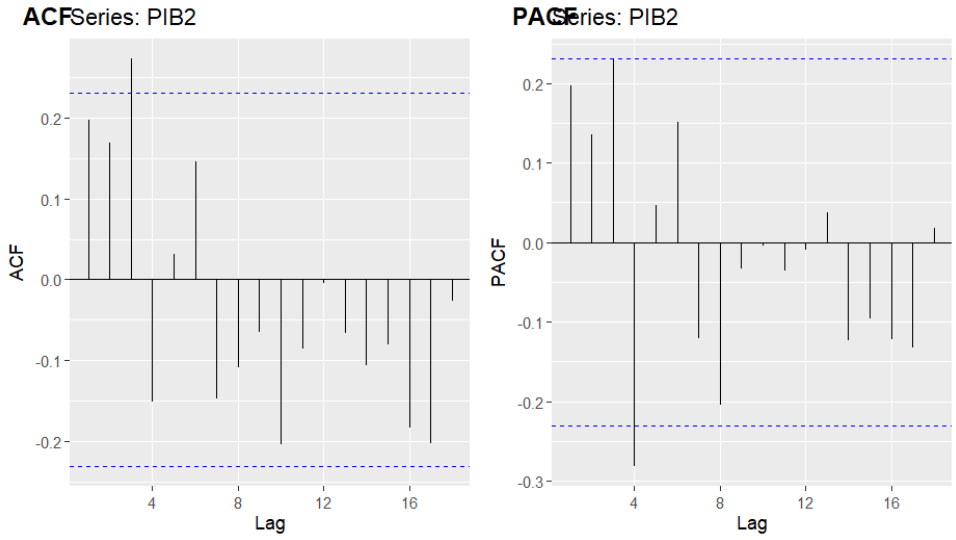

Figura 5: FAC y FACP del crecimiento del PIB de edificaciones

La Figura 5 muestra el resultado del modelo seleccionado por la función `auto.arima`. Este modelo no presenta resultados satisfactorios para el año 2023, pronosticando una tendencia divergente de la realidad observada con mayor o menor sensibilidad a la exclusión de los datos más antiguos. En contraste, el modelo ARIMA(3,0,1) muestra una mayor eficacia al capturar la tendencia futura de la serie.

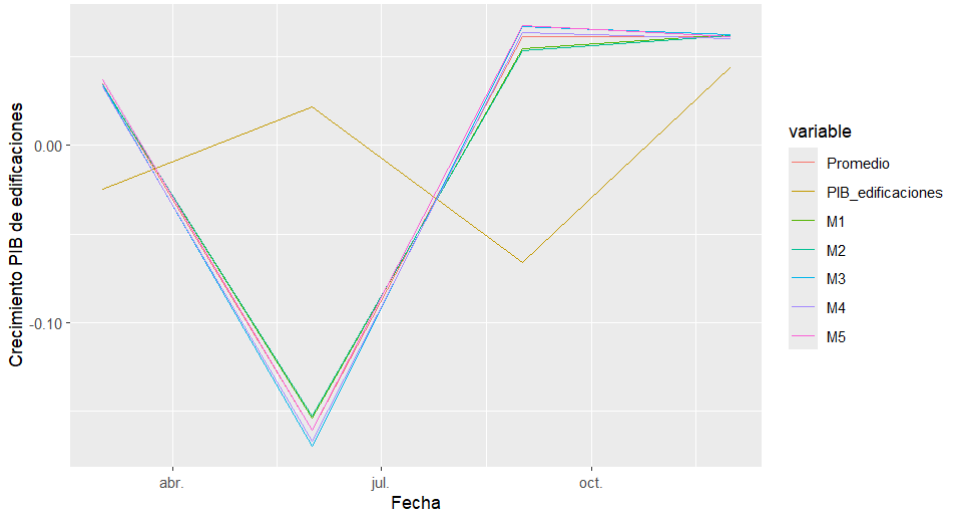

Figura 6: Pronósticos del PIB para el 2023 modelo ARIMA(1,0,1)(1,0,1)[4]

Nota: M_i representa la estimación del modelo ARIMA(1,0,1)(1,0,1)[4] eliminando las i observaciones más antiguas.

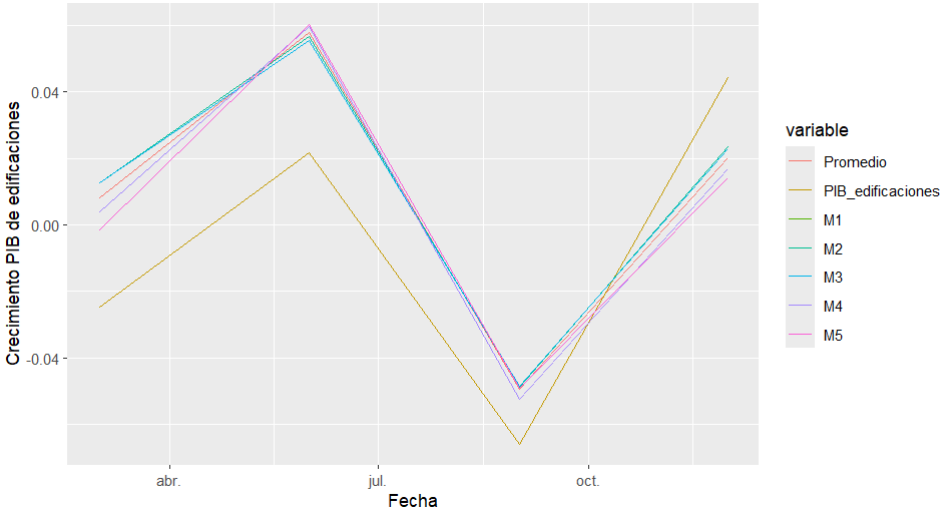

Figura 7: Pronósticos del PIB para el 2023 modelo ARIMA(3,0,1)

El desempeño de los tres modelos mejor clasificados se muestra en la Figura 8, junto con su promedio comparado con las verdaderas realizaciones de la serie durante 2023. La gráfica evidencia un buen comportamiento en la captura de la tendencia de la serie. Este resultado subraya la relevancia de los modelos ARIMA como complemento a otras metodologías avanzadas, integrando la solidez de los modelos estadísticos tradicionales con la sofisticación de técnicas modernas. Este enfoque híbrido asegura pronósticos más precisos y confiables, destacando su multifacética capacidad predictiva de manera particular en el PIB de edificaciones.

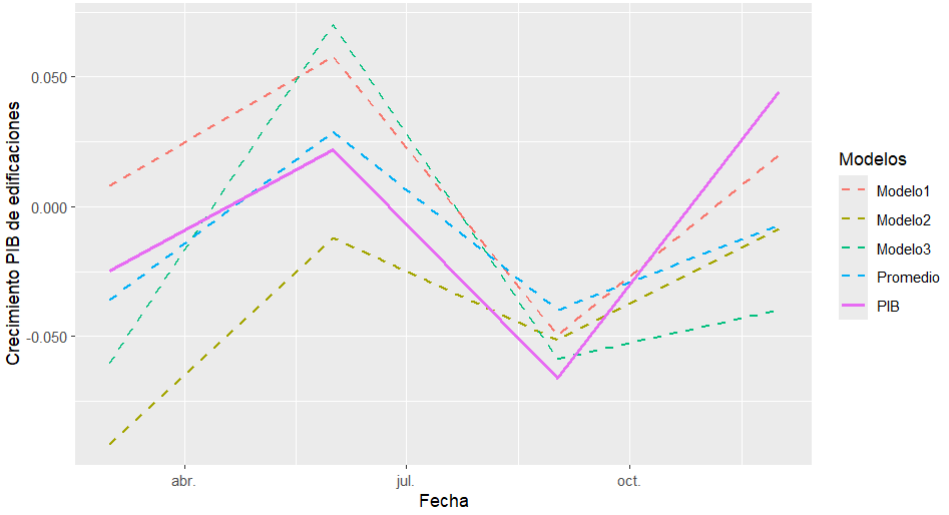

Figura 8: Pronosticos del PIB para 2023 modelos seleccionados versus su realización

Nota: Modelo1= ARIMA(3,0,1), Modelo2= ARIMA(2,0,4) y Modelo3= ARIMA(3,0,2).

El paso final en esta sección es determinar si los modelos seleccionados del enfoque ARIMA cumplen con los supuestos básicos de estacionariedad, no autocorrelación, normalidad y homocedasticidad. La Tabla 3 muestra que el modelo dos no supera tres de los cuatro supuestos planteados como básicos, el modelo tres tiene problemas de normalidad y homocedasticidad. Finalmente, el modelo uno registró solo problemas de heteroceasticidad. A pesar de esto se mantiene el uso de las especificaciones ecogidas bajo este enfoque, con el fin de comparar y aprovechar su capacidad para capturar la presencia de un atractor en la serie.

Modelo	Estacionariedad	Autocorrelación	Normalidad	Homocedasticidad
ARIMA(3,0,1)	Ok	Ok	Ok	Problemas p-valor=0.04
ARIMA(2,0,4)	Problemas p-valor=0.07	Ok	Problemas p-valor=0.006	Problemas p-valor=0.01
ARIMA(3,0,2)	Ok	Ok	Problemas p-valor=0.005	Problemas p-valor=0.01

Tabla 3: Verificación de supuestos en modelos ARIMA seleccionados

2.2. Enfoque multivariado

Para analizar las relaciones temporales, se propone estimar modelos de Vectores Autorregresivos (VAR). En su forma clásica, los VAR son una herramienta sencilla y flexible para capturar interacciones complejas entre variables macroeconómicas. Sin embargo, su potencial sobreparametrización puede ocasionar problemas, especialmente en la inferencia y en la realización de pronósticos. Para abordar esta cuestión, se consideran modelos VAR de tipo bayesiano; se debe aclarar que solo se toma el enfoque de Litterman(1986), que asume que las series analizadas son integradas de orden 1. Un modelo VAR de orden 1 con m variables puede representarse de la siguiente manera:

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t \tag{3}$$

Donde Y_t representa un vector de variables endógenas tomadas en el tiempo t dentro del intervalo $[1, T]$, y ε_t es un vector de choques exógenos distribuidos normalmente. El sistema descrito implica un total de $(k + pk^2)$ coeficientes para estimar. Este tamaño es considerable debido a la longitud promedio de las series utilizadas, lo que, a su vez, puede resultar en estimaciones con baja significancia, falta de precisión y problemas de correlación serial. En este contexto, este trabajo prueba el enfoque usado por (Thomas Doan & Sims 1984). Estos autores buscaron abordar estos inconvenientes al incluir información previa en las estimaciones, haciendo referencia a los posibles valores que podrían tomar los coeficientes, independientemente de la información derivada de los datos muestrales. Específicamente, se asume que las series macroeconómicas pueden describirse eficientemente mediante un proceso de caminata aleatoria de la forma $Y_{i,t} = \mu_t + Y_{i,t-1} + \varepsilon_{i,t}$ con $\varepsilon_{i,t} \sim N(0, \sigma^2)$.

En cuanto a la estructura sectorial para el enfoque multivariado, se utilizan los indicadores de la Tabla 4. Estas variables se seleccionaron de acuerdo con el esquema del proceso constructivo, el análisis gráfico y la definición de causación usada por el DANE para calcular el PIB de edificaciones (Departamento Administrativo Nacional de Estadística (DANE) 2022). La variable de formación de hogares y la tasa de interés hipotecaria para compra se incluyen como una forma de capturar el efecto de la demanda, mientras que las tasas para construcción se incluyen para tener en cuenta el efecto de los niveles de rentabilidad del negocio inmobiliario sobre los niveles de producción de valor agregado.

Tabla 4: Indicadores contenidos en el trabajo

Nombre	Fuente	Corte
Área iniciada	DANE	Microdato cuarto trimestre de 2024[a]
Área culminada	DANE	Microdato cuarto trimestre de 2024[a]
Área licenciada	DANE	Abril de 2024
Tasa de interés compra[b]	CAMACOL	Mayo de 2024
Tasa de interés constructor	CAMACOL	Mayo de 2024
PIB total [c]	DANE	Cuarto trimestre de 2023
PIB subsector de edificaciones	DANE	Cuarto trimestre de 2023
Área en proceso[d]	DANE	Microdato cuarto trimestre de 2024
Formación de hogares[e]	DANE	No admite

[a] Se toman los datos publicados por el DANE en:
"https://microdatos.dane.gov.co/index.php/catalog/construc-Microdatos"
[b] Tasa de colocación para créditos hipotecarios.
[c] Se toma la resta entre el PIB total y el PIB de edificaciones.

[d] Para la variable área en proceso, se toma la resta entre el área total en proceso reportada en el anexo DANE y el área iniciada en cada trimestre, información que puede ser consultada en el siguiente enlace: https://www.dane.gov.co/index.php/estadisticas-por-tema/construccion/censo-de-edificaciones.
[e] Esta información viene de forma anual, para efectos del documento se toma el valor de cada año y se divide en cuatro obteniendo de esta manera el valor de cada trimestre, esto permite controlar los efectos demográficos sobre la demanda.

Para determinar el número de rezagos apropiado, la inclusión de tendencia y constante, se utiliza un analisis gráfico, la función `VARselect` [6] de `R` y la prueba de raiz unitaria sobre el conjunto de las variables endógenas. La Figura 9 muestra que todas las series analizadas se mueven con mayor o menor volatilidad alrededor de cero; las pruebas de Dickey-Fuller y los criterios de información de la prueba `VARselect` muestran que la mejor opción es un modelo de entre dos y tres rezagos sin tendencia y constante. Sin embargo, se prueban algunas combinaciones con constante y tendencia para probar su eficacia dentro de la muestra.

La Figura 10 muestra la estimación de la dinámica del PIB (para 2023) de las especificaciones VAR seleccionadas, junto a su promedio. Estos resultados muestran un comportamiento menos eficiente que las especificaciones ARIMA seleccionadas en la sección anterior. Para validar los supuestos de normalidad, se utiliza la función `normality.test`, la cual realiza una prueba Jarque-Bera multivariada. Para determinar la existencia de correlación serial en los residuales del modelo, se utiliza la función `serial.test`. Finalmente, para el análisis de homocedasticidad, se utiliza la prueba multivariada de la función `arch.test`.

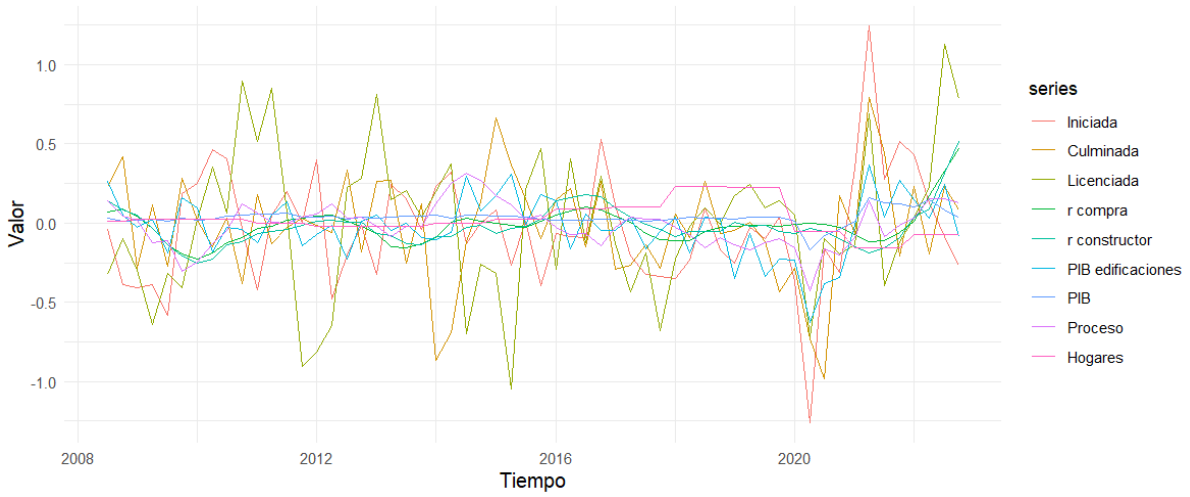

Figura 9: Evolición temporal de las variables incluidas

[6] este algoritmo determina el número óptimo de rezagos en un modelo VAR basado en los criterios de información: AIC, BIC, HQ y FPE

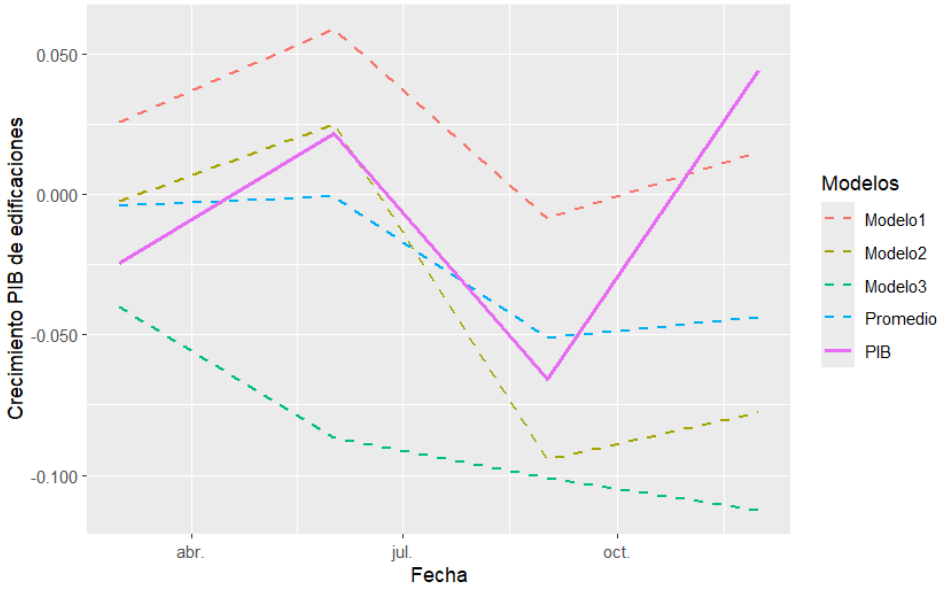

Figura 10: Pronosticos del PIB para 2023 modelos seleccionados versus su realización

Nota: Modelo1= VAR(2) constante, Modelo2= VAR(2) sin nada y Modelo3= VAR(1) sin nada.

Modelo	Estacionariedad	Autocorrelación	Normalidad	Homocedasticidad
VAR(2) con constante	Ok	Ok	Problemas	Ok
VAR(2) sin nada	Ok	Ok	Problemas	Ok
VAR(1) sin nada	Ok	Ok	Problemas	Ok

Tabla 5: Verificación de supuestos en modelos VAR seleccionados

2.2.1. Otras especifiaciones multivariadas

Ninguno de los modelos VAR especificados supera el supuesto de normalidad, lo cual es un problema ya que dificulta la inferencia estadística, genera estimadores no eficientes y, lo más importante, provoca poca fiabilidad en las predicciones. Con el fin de probar alternativas multivariadas, y para resolver los problemas de sobreajuste del enfoque VAR clásico, en la parte final de esta sección se realiza una exploración de los enfoques VARMA (Box et al. 1994) y VAR bayesiano (Thomas Doan & Sims 1984) ya mencionados en este documento.

Los modelos VARMA son una extensión multivariante de los modelos ARMA, y pueden ser representados como: $\mathbf{y}_t = \mathbf{c} + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} + \sum_{j=1}^{q} \mathbf{B}_j \mathbf{u}_{t-j} + \mathbf{u}_t$, donde $\mathbf{y}_t$ es un vector de variables endógenas en el tiempo t, $\mathbf{c}$ es un vector de constantes, $\mathbf{A}_i$ son matrices de coeficientes autorregresivos, $\mathbf{B}_j$ son matrices de coeficientes de media móvil, $\mathbf{u}_t$ es un vector de errores o innovaciones con media cero y matriz de covarianza $\mathbf{\Sigma}_u$. Los modelos VARMA son útiles para pronosticar el PIB debido a que al combinar componentes autorregresivos y de media móvil, pueden producir pronósticos que permitan ajustar tanto la tendencia de largo plazo como a las fluctuaciones a corto plazo. Debido a las exigencias computacionales, solo es posible obtener resultados para un VARMA(p=1,q=1), que incluye únicamente las variables PIB de edificaciones y el total del área en proceso. Esto impide aplicar el mismo procedimiento utilizado para las especificaciones VAR clásicas.

Al aplicar la metodología planteada en este documento en el enfoque bayesiano se tiene que las mejores especificaciones son: un VAR bayesiano de cuatro rezagos con distribución previa Minnesota, modelo que incluye área iniciada, en proceso, culminada y licenciada; un VAR bayesiano de tres rezagos con distribución previa Minnesota que incluye el área iniciada, en proceso, culminada y licenciada y un VAR bayesiano de tres rezagos con distribución previa Minnesota, modelo que incluye el total del área en proceso.

2.3. Redes neuronales

Como se mencionó al comienzo de este documento, este enfoque establece una analogía entre los estímulos recibidos por el cerebro y las variables independientes. El impacto de cada estímulo es determinado por el proceso de entrenamiento, que consiste en poner a prueba la capacidad de adaptación y autoorganización de la Red Neuronal Artificial (RNA) utilizando una base de datos donde los valores a estimar ya son conocidos. Esto permite contrastar y calibrar el modelo mediante medidas basadas en el error.

La Figura 11 muestra la forma más común de representar una Red Neuronal Artificial (RNA). En esta representación, la primera sección de la red neuronal (en color verde) se conoce como capa de entrada y recibe los datos en bruto, es decir, los valores de los predictores. La parte intermedia (en color azul), conocida como capa oculta, recibe los valores de la capa de entrada ponderados por los pesos (flechas grises). Finalmente, se encuentra la denominada capa de salida, que combina los valores provenientes de la capa intermedia y genera la predicción. Dentro de la unidad fundamental de la RNA solo ocurren dos operaciones: la suma ponderada de sus entradas y la aplicación de una función de activación.

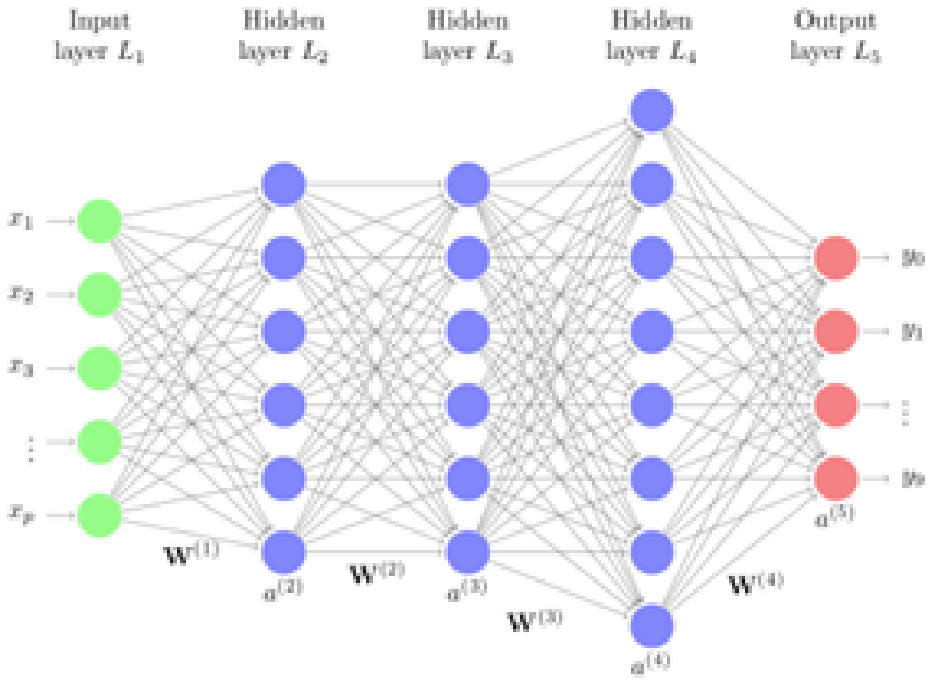

Figura 11: Modelo RNA de varias capas ocultas

A continuación, se describen los componentes clave del proceso de estimación de una RNA.

2.3.1. Función de activación

La función de activación controla en gran medida qué información se propaga desde una capa a la siguiente. Esta convierte el valor neto de entrada a la neurona en un nuevo valor. Gracias a la combinación de funciones de activación no lineales con múltiples capas ocultas, los modelos de redes son capaces de aprender relaciones no lineales. La gran mayoría convierten el valor de entrada neto de la neurona en un valor dentro del rango (0, 1) o (-1, 1). Cuando el valor de activación de una neurona es cero, se dice que la neurona está inactiva, ya que no pasa ningún tipo de información a las siguientes neuronas. A continuación, se describen las funciones de activación más empleadas.

+ Rectified linear unit (ReLU): Esta función aplica una transformación no lineal que activa la neurona solo si la entrada está por encima de cero. De tal manera que $ReLU(x) = \max(x, 0)$. ReLU es la función más empleada por sus buenos resultados en aplicaciones diversas y simplicidad.

+ Sigmoide: La función sigmoide transforma valores en el rango de $(-\infty, \infty)$ a valores en el rango $(0, 1)$. Aunque esta función se utilizó en los inicios de los modelos de redes, en la actualidad, suele preferirse la función ReLU. Un caso en el que esta función de activación sigue siendo la utilizada por defecto es en las neuronas de la capa de salida de los modelos de clasificación binaria, ya que su salida puede interpretarse como probabilidad.

+ Tangente hiperbólica (Tanh): La función de activación Tanh, se comporta de forma similar a la función sigmoide, pero su salida está acotada en el rango (-1, 1). La transformación Tanh viene dada por $\frac{1-\exp^{-2x}}{1+\exp^{2x}}$

2.3.2. Función de costo

La función de costo, también llamada función de pérdida, es la encargada de cuantificar la distancia entre el valor real y el valor predicho por la red. En otras palabras, mide cuánto se equivoca el modelo al realizar predicciones. La función de costo puede calcularse para una única observación o para un conjunto de datos. Es este último caso el que se utiliza para dirigir el entrenamiento de los modelos. Dependiendo del tipo de problema, ya sea regresión o clasificación, es necesario utilizar una función de costo específica. En problemas de regresión, las más utilizadas son el error cuadrático medio y el error absoluto medio. En problemas de clasificación, suele emplearse la función de pérdida logarítmica o log-loss, definida como:

$$\text{Log-loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Donde: N es el número de observaciones, y_i es la etiqueta verdadera (0 o 1) para la i-ésima observación y $\hat{y}_i$ es la probabilidad predicha de que la i-ésima observación pertenezca a la clase positiva (1).

2.3.3. Resultados redes neuronales

Para la aplicación del enfoque de redes neuronales, se exploran especificaciones que contienen tres rezagos, basándose en los resultados obtenidos con las metodologías VAR y ARIMA. En cuanto a la configuración de la red neuronal, se utiliza un máximo de diez neuronas por capa y se emplean hasta tres capas ocultas.

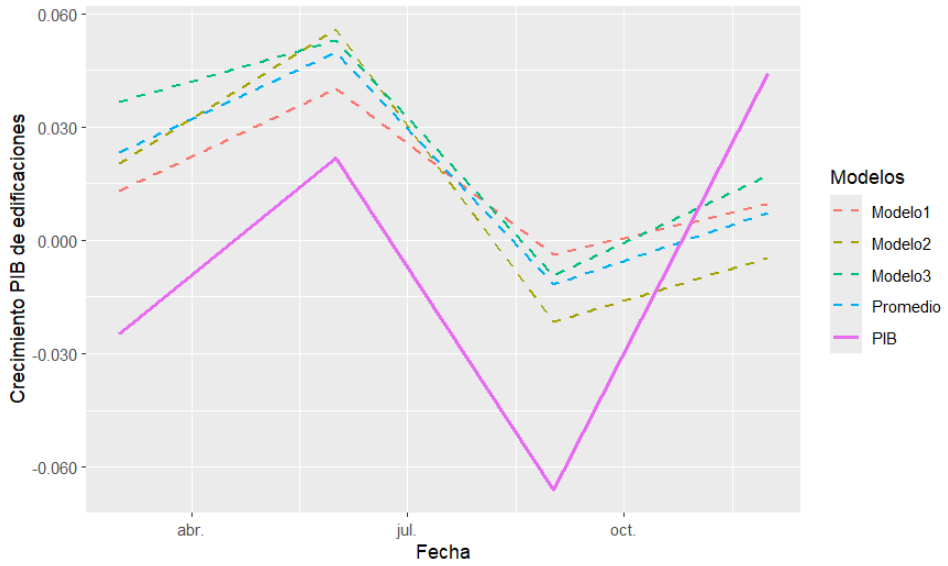

Figura 12: Pronosticos del PIB para 2023 modelos seleccionados RNA versus su realización

Nota: `Modelo1= tres rezagos, dos capas ocultas y cinco neurona por capa; Modelo2= tres rezagos, dos capas ocultas y ocho neurona por capa y Modelo3= tres rezagos, dos capas ocultas y una neurona por capa.`

Después de evaluar el desempeño de las redes entrenadas frente a la dinámica presentada por el PIB de edificaciones durante 2023, se seleccionaron las siguientes especificaciones:

- Modelo1: tres rezagos, dos capas ocultas y cinco neuronas por capa. Los resultados de esta especificación están en la Figura 13.
- Modelo2: tres rezagos, dos capas ocultas y ocho neuronas por capa. Los resultados de esta especificación están en la Figura 14.
- Modelo3: tres rezagos, dos capas ocultas y una neurona por capa. Los resultados de esta especificación están en la Figura 15.

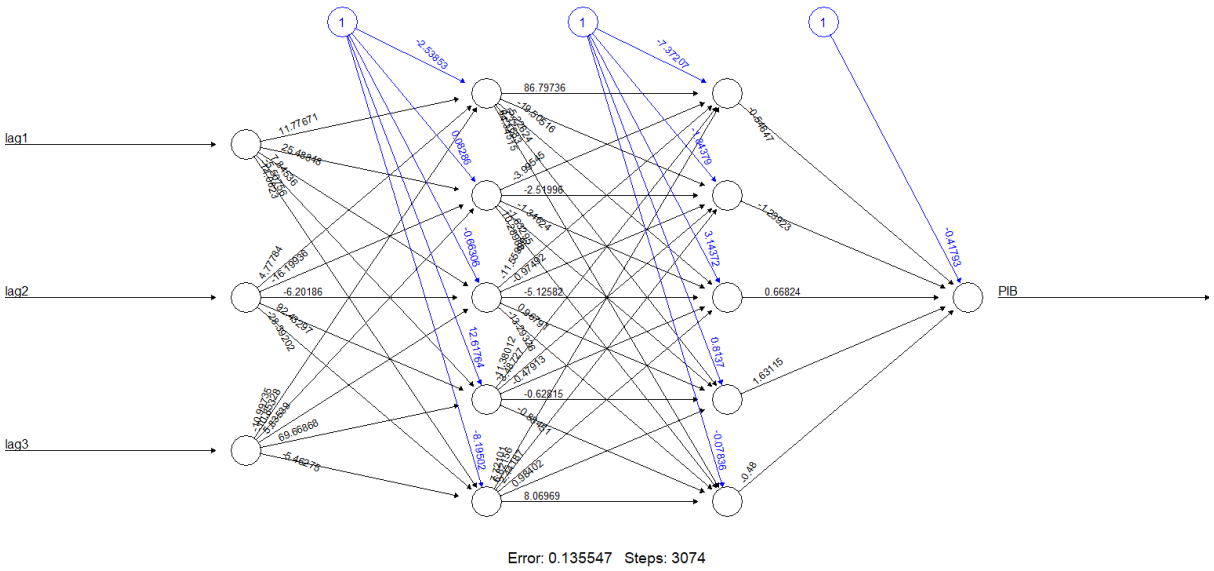

Figura 13: Modelo RNA tres rezagos dos capas ocultas y cinco neuronas por capa

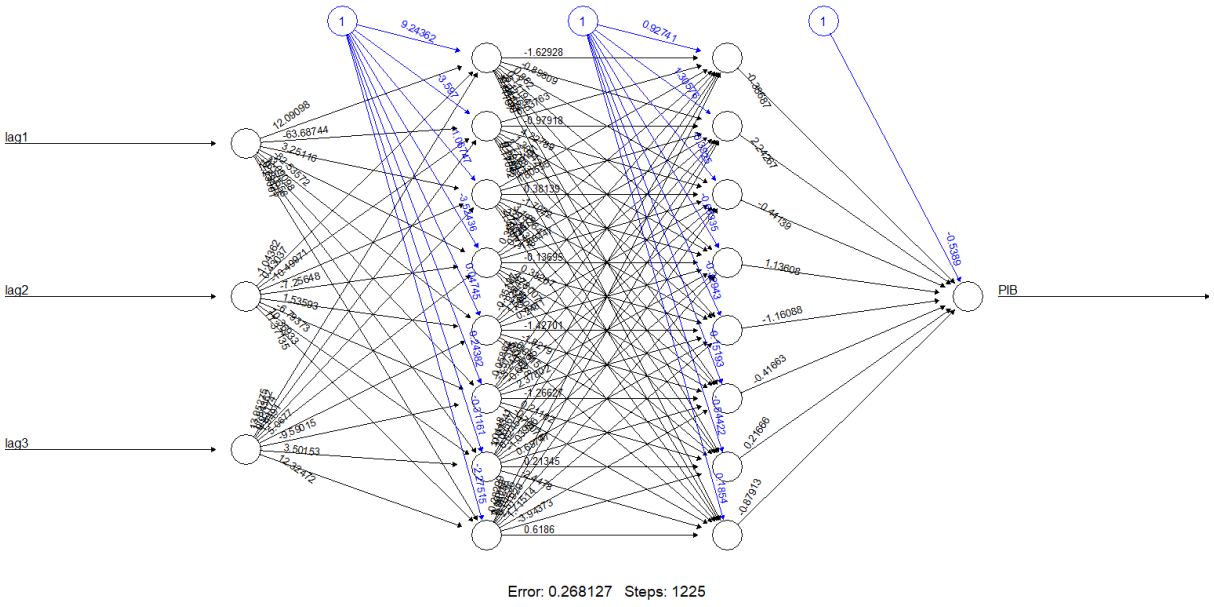

Figura 14: Modelo RNA tres rezagos dos capas ocultas y ocho neuronas por capa

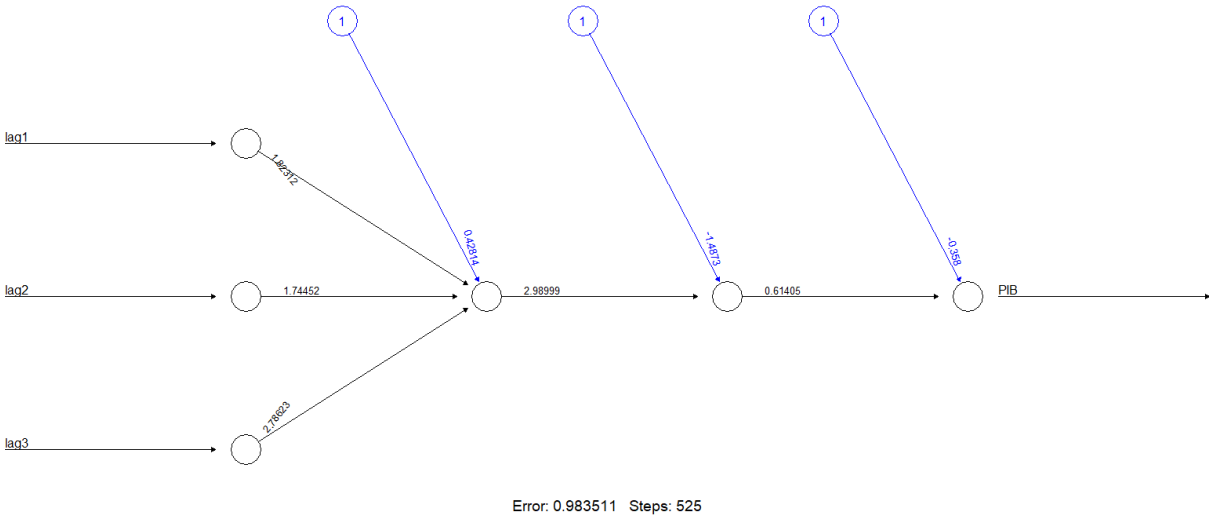

Figura 15: Modelo RNA tres rezagos dos capas ocultas y una neurona por capa

3. Resultados

En esta sección se presentan los pronósticos para 2024 y 2025 de las mejores especificaciones. En la tabla 6 se muestran los resultados de los modelos seleccionados en este trabajo, utilizando el indicador combinado para clasificar cada enfoque. Finalmente, en la gráfica se presentan los pronósticos, incluyendo los escenarios optimista y conservador. El escenario medio está dado por el promedio de las proyecciones realizadas por las herramientas seleccionadas, el escenario conservador se define como el promedio menos dos veces la desviación estándar del conjunto de proyecciones, y el escenario optimista se define como el promedio más dos veces la desviación estándar de las proyecciones.

Modelo	**Índice de desempeño**	**I-2024**	**II-2024**	**III-2024**	**IV-2024**	**I-2025**	**II-2025**	**III-2025**	**IV-2025**
ARIMA(3,0,1)	-1.27	991	990	1.069	1.123	986	988	1.057	1.118
ARIMA(2,0,4)	-0.86	1.019	970	1.077	1.137	986	981	1.069	1.114
ARIMA(3,0,2)	-0.77	1.046	985	1.059	1.162	1.016	984	1.068	1.133
VAR(2) con constante	-0.78	1.115	1.185	1.277	1.322	1.210	1.182	1.163	1.119
VAR(2) sin parte deterministica	-0.75	1.107	1.170	1.228	1.277	1.167	1.140	1.110	1.091
VAR(1) sin parte deterministica	-0.51	1.130	1.095	1.104	1.148	1.105	1.050	1.050	1.089
VAR bayesiano Minnesota(4)	-1.06	1.159	1.162	1.183	1.206	1.216	1.222	1.226	1.229
VAR bayesiano Minnesota(3)	-0.97	1.168	1.191	1.214	1.228	1.240	1.249	1.255	1.261
VAR bayesiano Minnesota(3)[7]	-0.74	1.161	1.194	1.224	1.234	1.240	1.250	1.255	1.259
RNA de tres rezagos dos capas ocultas y cinco neuronas por capa	-0.64	1.031	884	1.204	1.214	913	991	1.301	1.066
RNA de tres rezagos dos capas ocultas y ocho neuronas por capa	-0.62	971	1.021	971	1.185	970	1.014	907	1.207
RNA de tres rezagos dos capas ocultas y una neurona por capa	-0.61	1.069	1.068	1.128	1.186	1.147	1.131	1.208	1.232

Tabla 6: Resultados finales

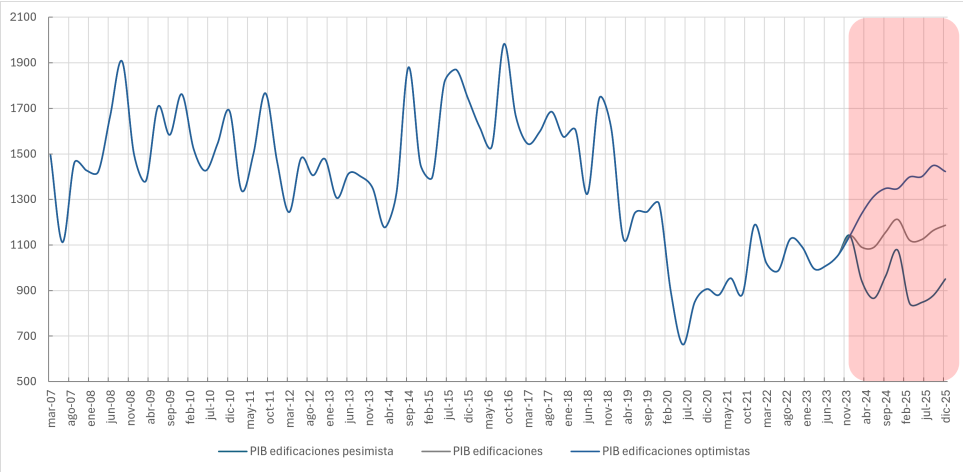

Figura 16: Pronóstico PIB en precios constantes de 2015

4. Conclusiones

Este ejercicio muestra que los enfoques analizados aportan información relevante sobre el posible comportamiento futuro de la actividad edificadora en la ciudad, bajo las limitaciones por falta de volumen de datos a que se enfrenta el enfoque de redes neuronales a la hora de hacer pronósticos puntuales. El enfoque no puede desplegar todo su potencial dada la poca disponibilidad de escenarios de aprendizaje; sin embargo su principal aporte se observa al modelar la tendencia de corto plazo. En cuanto al enfoque multivariado clásico, parece describir bastante bien la tendencia de la serie, debido a la similitud de funcionamiento que tiene con el proceso de causación usado por el DANE para generar el PIB. Es importante aclarar que este documento no agota todas las posibilidades de cada metodología como herramienta de pronóstico y qee es solo un trabajo exploratorio que busca aportar en el tema que está limitado por la información disponible y las limitaciones técnicas.

Con el fin de evaluar la presición en el día a día de los cálculos realizados, se compara el 9,5 % estimado por la metodología para el primer trimestre de 2024 frente al valor reportado en agosto de 2024 (10,2 %) por el DANE, lo cual si bien es un muy buen resultado esta metología debe estar en constante mejora y actualización.

Recibido:
Aceptado:

Referencias

Bovas, A. & Thavaneswaran, A. (1991), 'A nonlinear time series model and estimation of missing data.', *Annals of the Institute of Statistical Mathematics* **43**(3), 493–504.

Box, G. E., Jenkins, G. M. & Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control*, Prentice Hall.

Box, G. E. P. & Jenkins, G. M. (1970), *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.

Departamento Administrativo Nacional de Estadística (DANE) (2022), 'Nota metodológica: Implementación de la fase 2 del pib'. Accedido: 2024-07-13.
*https://www.dane.gov.co/files/investigaciones/boletines/pib/pib-nota-metodologica-implementacion-fase2-15-feb-2022.pdf

EnvStats Development Team (2023), *EnvStats: An R Package for Environmental Statistics*. Versión 2.5.1.
*https://CRAN.R-project.org/package=EnvStats

Hyndman, R. J. & Athanasopoulos, G. (2018), *Forecasting: Principles and Practice*, OTexts, Melbourne.

R Core Team (2023), *urca: Unit Root and Cointegration Tests for Time Series Data*, R Foundation for Statistical Computing, Vienna, Austria.
*https://cran.r-project.org/web/packages/urca/index.html

Thomas Doan, R. L. & Sims, C. (1984), 'Forecasting and conditional projection using realistic prior distributions', *Econometric Reviews* **3**(1), 1–100.
*https://doi.org/10.1080/07474938408800053

Wilcox, R. R. (2012), *Introduction to Robust Estimation and Hypothesis Testing*, Academic Press, San Diego.

A. Códigos

```
# ------------------------------------------------------------------------------
# ------------------------------------------------------------------------------
# -----------------Programacion pronostico subsector edificador-----------------
# ------------------------------------------------------------------------------
# ------------------------------------------------------------------------------

# Objetivo: evaluar el desempeño de los enfoques univarido y multivarido para estimar
# el comportamiento futuro del PIB de edificaciones de la ciudad de Bogotá y
# seleccionar un conjunto optimo de especificaciones para ser usadas posteriormente.
# Autores: Cristian Andrés Torres Casallas y Daniela Sedano Saenz
# Fecha de actualizacion: 2024-07-27

# ------------------------------------------------------------------------------
# ------------------------------Librerias necesarias----------------------------
# ------------------------------------------------------------------------------
# Las lineas que contienen la función install.packages deben ser corridas solo una
# vez y estan encargadas de descargar las funciones necesarias para que la programacion
# funcione.
 install.packages("readxl")
 install.packages("rlang")
 install.packages("pillar")
 install.packages("tables")
 install.packages("dplyr")
 install.packages("formattable")
 install.packages("tidyr")
 install.packages("ggplot2")
 install.packages("vars")
 install.packages("forecast")
 install.packages("ggpmisc")
 install.packages("gtable")
 install.packages("outliers")
 install.packages("EnvStats")
 install.packages("car")
 install.packages("reshape2")
 install.packages("reshape")
 install.packages("dCovTS")
 install.packages("datawizard")
 install.packages("bsts")
 install.packages("BoomSpikeSlab")
 install.packages("bayesforecast")
 install.packages("EnvStats")
 install.packages("dCovTS")
 install.packages("datawizard")
```

```
 install.packages("MTS")
 install.packages("ggpubr")

 library("ggpubr")
library("reshape2")
library("bayesforecast")
library("lubridate")
library("bsts")
library("dplyr")
library("ggplot2")
library("EnvStats")
library("readxl")
library("rlang")
library("pillar")
library("dplyr")
library("tidyr")
library("lattice")
library("vars")
library("forecast")
library("ggpmisc")
library("urca")
library("outliers")
library("car")
library("dCovTS")
library("datawizard")
library(vars)
library(lmtest)
library(tseries)
library(urca)


# ------------------------------------------------------------------------------
# -------------------------------Modelos univariados----------------------------
# ------------------------------------------------------------------------------
# Primero se toma el enfoque univariado, el cual incluye Arimas de enfoque clasicos.

# -------------------------------Llamado de los datos---------------------------
# Para los calculos se toma la diferencia anual del logaritmo natural del PIB de
# edificaciones, la cual es una buena aproximación del crecimeitno anual y empiricamente
# ha mostrado buenos resultados.
# Se eliminan todos los objetos activos en R.
rm(list=ls(all=TRUE))

# Se cambia el directorio de trabajo, este debe coincidir con la carpeta de los datos
setwd("D:/Articulo redes/Bases y programaciones/")

# Se importa la serie completa de PIB, la cual se calcula desde 2005
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "G1:G77",sheet ="Base")),frequency = 4, start = c(2


# ---------------------------------Datos atipicos-------------------------------
# --Boxplot para mirar atipicos
Base<-data.frame(PIB,Trimestre=c(rep(c("I","II","III","IV"),18)))
Boxplot(PIB_edificaciones ~ Trimestre, data = Base)

# --Prueba sobre todos los puntos de la serie
test <- rosnerTest(PIB,k = 3)
test
```

```
# --Graficas de los atipicos de la serie
s <- as.Date("2006-03-01")
e <- as.Date("2023-12-01")
Fecha<-as.Date(seq(from=s, to=e, by="3 months"))
PIB1<- diff(ts(log(read_excel("Base.xlsx",range = "G1:G77",sheet ="Base")),frequency = 4, start = c(
PIB2<-diff(ts(log(read_excel("Base.xlsx",range = "G1:G77",sheet ="Base")),frequency = 4, start = c(2
Media<-Base%>%group_by(Trimestre)%>%
  summarise_all(mean)
a<-Media$PIB_edificaciones[2]
b<-Media$PIB_edificaciones[1]
c<-Media$PIB_edificaciones[4]

PIB2[58]<-a

df1<-data.frame(Fecha=Fecha,Edificaciones=PIB1,Edificaciones_Ajustado=PIB2)
df<-melt(df1,id=c("Fecha"))
shade <- data.frame(x1 = c(as.Date("2020-03-01")),
                    x2 = c(as.Date("2020-12-01")),
                    min = c(-Inf), max = c(Inf))
ggplot() +
geom_line(data = df, aes(x = Fecha, y = value,color = variable))+
geom_rect(data = shade, aes(xmin = x1, xmax = x2, ymin = min, ymax = max),
            fill = c("blue"), alpha = 0.2)


#-----------------------------Estadisticas de resumen---------------------------
mean(PIB2)
median(PIB2)
max(PIB2)
min(PIB2)
sd(PIB2)
sd(PIB2)/mean(PIB2)

#-----------------------------Pruebas de raiz unitaria--------------------------

#Prueba Dickey-Fuller, no se acepta la hipotesis nula, con lo cual es adecuado
#utilizr la metodologia de Box-jenkins
summary(ur.df(PIB, type = "trend", lags = 0))
summary(ur.df(PIB, type = "trend", lags = 1))
summary(ur.df(PIB, type = "trend", lags = 2))

summary(ur.df(PIB, type = "drift", lags = 0))
summary(ur.df(PIB, type = "drift", lags = 1))
summary(ur.df(PIB, type = "drift", lags = 2))

summary(ur.df(PIB, type = "none", lags = 0))
summary(ur.df(PIB, type = "none", lags = 1))
summary(ur.df(PIB, type = "none", lags = 2))


#--------------------Graficas para identificar un posible modelo----------------

acf_plot <- ggAcf(PIB2, lag.max = NULL, type = "correlation", plot = T, na.action = na.contiguous, d
pacf_plot <- ggAcf(PIB2, lag.max = NULL, type = "partial", plot = T, na.action = na.contiguous, deme

ggarrange(acf_plot, pacf_plot, ncol = 2,
          labels = c("ACF", "PACF"),
```

```
          common.legend = TRUE, legend = "bottom")

#------------------------Estimación univariados ARIMA----------------------------
#Se estiman varios modelos bajo el enfoque Box-Jenkins y se evalua su desempeño
#segmentando la serie y creando pronosticos dentro de muestra.
PIB<-window(PIB2,start=c(2006,1),end=c(2022,4))
Prueba<-window(PIB2,start=c(2023,1),end=c(2023,4))
auto.arima(PIB)
#--Generamos los parametros iniciales del ejericico
#n:Tamaño de la serie con la cual se estiman los modelos
#h:numero de pasos adelante que deben pronosticar los modelos
#k:numero maximo de datos excluidos de la cola de la serie
n<-length(PIB)
h<-length(Prueba)
k<-5

#--Modelo 1 (función auto arima)
#Se construye la matriz para rellenar con los pronosticos
P0<-matrix(0, nrow = h, ncol =k )
#Iteracion para generar los pronosticos y rellenar la matriz anterior
for (i in 1:k) {
P0[,i]<-predict(arima(PIB[i:n], order=c(1, 0, 1),seasonal=list(order=c(1,0,1), period=4)), h)$pred[1
}

#Se evaluan aquellos modelos que no generaron correctamente las
#estimaciones
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P0==0)>porcentaje)
#Se genera el promedio para las distintas estimaciones
if (length(columnas_a_borrar)>0){
P0 <- matrix(as.numeric(P0[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico0<-rowSums(P0)/(k-length(columnas_a_borrar))} else {Pronostico0<-rowSums(P0)/k}
#Generar la graficas
s <- as.Date("2023-03-01")
e <- as.Date("2023-12-01")
Fecha<-as.Date(seq(from=s, to=e, by="3 months"))
Grafico<-data.frame(Promedio=Pronostico0,Prueba,M1=P0[,1],M2=P0[,2],M3=P0[,3],M4=P0[,4],M5=P0[,5],Fe
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))

#--Modelo 2
P01<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
P01[,i]<-predict(arima(PIB[i:n], order=c(3, 0, 4),seasonal =list(order=c(1,0,1), period=4)), h)$pred
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P01==0)>porcentaje)
if (length(columnas_a_borrar)>0){
P01 <- matrix(as.numeric(P01[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico01<-rowSums(P01)/(k-length(columnas_a_borrar))} else {Pronostico01<-rowSums(P01)/k}

Grafico<-data.frame(Promedio=Pronostico01,Prueba,M1=P01[,1],M2=P01[,2],M3=P01[,3],M4=P01[,4],M5=P01[
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
```

```
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))


#--Modelo 3
P02<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  P02[,i]<-predict(arima(PIB[i:n], order=c(3, 0, 4)), h)$pred[1:h]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P01==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P02 <- matrix(as.numeric(P02[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico02<-rowSums(P02)/(k-length(columnas_a_borrar))} else {Pronostico02<-rowSums(P02)/k}

Grafico<-data.frame(Promedio=Pronostico02,Prueba,M1=P02[,1],M2=P02[,2],M3=P02[,3],M4=P02[,4],M5=P02[
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))

#--Modelo 4
P03<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  P03[,i]<-predict(arima(PIB[i:n], order=c(3, 0, 3)), h)$pred[1:h]
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P03==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P03 <- matrix(as.numeric(P03[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico03<-rowSums(P03)/(k-length(columnas_a_borrar))} else {Pronostico03<-rowSums(P03)/k}

Grafico<-data.frame(Promedio=Pronostico03,Prueba,M1=P03[,1],M2=P03[,2],M3=P03[,3],M4=P03[,4],M5=P03[
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))

#--Modelo 5
P04<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  P04[,i]<-predict(arima(PIB[i:n], order=c(3, 0, 2)), h)$pred[1:h]
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P04==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P04 <- matrix(as.numeric(P04[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico04<-rowSums(P04)/(k-length(columnas_a_borrar))} else {Pronostico04<-rowSums(P04)/k}

Grafico<-data.frame(Promedio=Pronostico04,Prueba,M1=P04[,1],M2=P04[,2],M3=P04[,3],M4=P04[,4],M5=P04[
df<-melt(Grafico,id=c("Fecha"))
```

```
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))

#--Modelo 6
P05<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  P05[,i]<-predict(arima(PIB[i:n], order=c(3, 0, 1)), h)$pred[1:h]
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P05==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P05 <- matrix(as.numeric(P05[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico05<-rowSums(P05)/(k-length(columnas_a_borrar))} else {Pronostico05<-rowSums(P05)/k}

Grafico<-data.frame(Promedio=Pronostico05,Prueba,M1=P05[,1],M2=P05[,2],M3=P05[,3],M4=P05[,4],M5=P05[
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))

#--Modelo 7
P06<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  P06[,i]<-predict(arima(PIB[i:n], order=c(2, 0, 4)), h)$pred[1:h]
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P06==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P06 <- matrix(as.numeric(P06[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico06<-rowSums(P06)/(k-length(columnas_a_borrar))} else {Pronostico06<-rowSums(P06)/k}

Grafico<-data.frame(Promedio=Pronostico06,Prueba,M1=P06[,1],M2=P06[,2],M3=P06[,3],M4=P06[,4],M5=P06[
df<-melt(Grafico,id=c("Fecha"))
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line() +
  xlab("Fecha") +   # Etiqueta para el eje x
  ylab("Crecimiento PIB de edificaciones") +  # Etiqueta para el eje y sin formato adicional
  scale_y_continuous(labels = scales::number_format(accuracy = 0.01))


Resutado1<-data.frame(Pronostico0,Pronostico01,Pronostico02,Pronostico03,Pronostico04,
Pronostico05,Pronostico06)

#----------------------------------Selección de los modelos--------------------
z<-3
RMS<-(colSums((Resutado1-c(Prueba))*(Resutado1-c(Prueba))))/length(Prueba)
AMS<-(colSums(abs(Resutado1-c(Prueba))))/length(Prueba)
Promedio<-data.frame(Prome=(scale(RMS)+scale(AMS))/2)

orden_filas <- order(Promedio$Prome)
Modelos <- Promedio[orden_filas, ]
Modelos <- data.frame(row.names=c(rownames(Promedio)[orden_filas]),Promedio=Modelos)
Modelos_selec_ARIMA<-rownames(Modelos)[1:z]
```

```
#---------------------------Construcción del pronostico conjunto ARIMA----------
#Promedio normal
Arima<-Resutado1[,Modelos_selec_ARIMA]
Arima$Promedio_Arima<-(rowSums(Arima))/3


s <- as.Date("2023-03-01")
e <- as.Date("2023-12-01")
Fecha <- as.Date(seq(from = s, to = e, by = "3 months"))

Prueba<-data.frame(Prueba)
colnames(Prueba)<-"PIB"
Grafico <- data.frame(  "Modelo1" = Arima[, 1],  "Modelo2" = Arima[, 2],  "Modelo3" = Arima[, 3],  P
  PIB=Prueba,
  Fecha)

df <- melt(Grafico, id = c("Fecha"))

modelo_resaltado <- "PIB"

ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line(size = 0.8, linetype = "dashed") +  # Todas las líneas
  geom_line(data = subset(df, variable == modelo_resaltado), size = 1) +  # Línea resaltada
  xlab("Fecha") +
  ylab("Crecimiento PIB de edificaciones") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.005)) +
  labs(color = "Modelos")  # Etiqueta para la leyenda

#-------------------------------Validación de supuestos-------------------------
library(tseries)
library(lmtest)
PIB<-PIB2
#---- Modelos 1 ARMA (3,0,1)
Modelo_1<-arima(PIB, order=c(3, 0, 1))

# 1. Verificar graficamente los residuos
checkresiduals(Modelo_1)
adf_test <- adf.test(Modelo_1$residuals) #H_0: no estacionariedad
print(adf_test)

# 2. Prueba de Ljung-Box para autocorrelación H_0=no autocorrelación
ljung_box_test <- Box.test(Modelo_1$residuals, lag=10, type="Ljung-Box")
print(ljung_box_test)

# 3. Verificar Normalidad de los residuos H_0=normalidad
shapiro_test <- shapiro.test(Modelo_1$residuals)
print(shapiro_test)

# 6. Verificar Homoscedasticidad H_0=homoceasticidad
bptest_result <- bptest(Modelo_1$residuals ~ fitted(Modelo_1))
print(bptest_result)


#---- Modelos 2 ARMA (2,0,4)
Modelo_2<-arima(PIB, order=c(2, 0, 4))

# 1. Verificar graficamente los residuos
```

```
checkresiduals(Modelo_2)
adf_test <- adf.test(Modelo_2$residuals) #H_0: no estacionariedad
print(adf_test)

# 2. Prueba de Ljung-Box para autocorrelación H_0=no autocorrelación
ljung_box_test <- Box.test(Modelo_2$residuals, lag=10, type="Ljung-Box")
print(ljung_box_test)

# 3. Verificar Normalidad de los residuos H_0=normalidad
shapiro_test <- shapiro.test(Modelo_2$residuals)
print(shapiro_test)

# 6. Verificar Homoscedasticidad H_0=homoceasticidad
bptest_result <- bptest(Modelo_2$residuals ~ fitted(Modelo_1))
print(bptest_result)

#---- Modelos 3 ARMA (3,0,2)
Modelo_3<-arima(PIB, order=c(3, 0, 2))

# 1. Verificar graficamente los residuos
checkresiduals(Modelo_3)
adf_test <- adf.test(Modelo_3$residuals) #H_0: no estacionariedad
print(adf_test)

# 2. Prueba de Ljung-Box para autocorrelación H_0=no autocorrelación
ljung_box_test <- Box.test(Modelo_3$residuals, lag=10, type="Ljung-Box")
print(ljung_box_test)

# 3. Verificar Normalidad de los residuos H_0=normalidad
shapiro_test <- shapiro.test(Modelo_3$residuals)
print(shapiro_test)

# 6. Verificar Homoscedasticidad H_0=homoceasticidad
bptest_result <- bptest(Modelo_3$residuals ~ fitted(Modelo_1))
print(bptest_result)

#---- Modelos 4 ARMA (3,0,3)
Modelo_4<-arima(PIB, order=c(3, 0, 3))

# 1. Verificar graficamente los residuos
checkresiduals(Modelo_4)
adf_test <- adf.test(Modelo_4$residuals) #H_0: no estacionariedad
print(adf_test)

# 2. Prueba de Ljung-Box para autocorrelación H_0=no autocorrelación
ljung_box_test <- Box.test(Modelo_4$residuals, lag=10, type="Ljung-Box")
print(ljung_box_test)

# 3. Verificar Normalidad de los residuos H_0=normalidad
shapiro_test <- shapiro.test(Modelo_4$residuals)
print(shapiro_test)

# 6. Verificar Homoscedasticidad H_0=homoceasticidad
bptest_result <- bptest(Modelo_4$residuals ~ fitted(Modelo_1))
print(bptest_result)

# ------------------------------------------------------------------------------
# ------------------------------Modelos multivariados---------------------------
```

```
# ------------------------------------------------------------------------------
# Aplcación de los modelos VAR

# Importación de los datos
rm(list=ls(all=TRUE))
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "B1:J77",sheet ="Base")),frequency = 4, start = c(2
Prueba<-window(PIB[,6],start=c(2023,1),end=c(2023,4))
PIB<-na.omit(window(PIB, start = c(2006, 1), end = c(2022, 4)))

#---- Parametros iniciales

# Horizonte de tiempo para pronosticar
h<-length(Prueba)
# Maximo numero de datos a eliminar de la base en las estimaciones
k<-5
# Definir el numero de rezagos
# Primero definimos si el modelo necesita constante o tendencia
colnames(PIB)[1]<-"Iniciada"
colnames(PIB)[2]<-"Culminada"
colnames(PIB)[3]<-"Licenciada"
colnames(PIB)[4]<-"r compra"
colnames(PIB)[5]<-"r constructor"
colnames(PIB)[6]<-"PIB edificaciones"
colnames(PIB)[7]<-"PIB"
colnames(PIB)[8]<-"Proceso"
colnames(PIB)[9]<-"Hogares"
# Grafico
autoplot(PIB) +  ggtitle("") +  xlab("Tiempo") +  ylab("Valor") +  theme_minimal() +
  theme(    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),    axis.title.x = elem
# Ajustar decimales
options(scipen = 999)
# Encontrar la mejor especificacion
Solo<-VARselect(PIB, lag.max = 3, type = "none")
Constante<-VARselect(PIB, lag.max = 3, type = "const")
Tendencia<-VARselect(PIB, lag.max = 3, type = "trend")
Ambos<-VARselect(PIB, lag.max = 3, type = "both")

# Gana el modelo sin nada ya que tiene el AIC y BIC más pequeño

#---- Estimación
# Mdelo 1 VAR(3)
P1<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 3,type ="none")
if (max(roots(p1ct))<1)
{P1[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
else {P1[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P1==0)>porcentaje)
if (length(columnas_a_borrar)>0){
P1 <- matrix(as.numeric(P1[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico1<-rowSums(P1)/(k-length(columnas_a_borrar))} else {Pronostico1<-rowSums(P1)/k}

# Mdelo 2 VAR(2) sin nada
P2<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
```

```
  p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 2,type ="none")
  if (max(roots(p1ct))<1)
  {P2[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
  else {P2[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P2==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P2 <- matrix(as.numeric(P2[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico2<-rowSums(P2)/(k-length(columnas_a_borrar))} else {Pronostico2<-rowSums(P2)/k}

# Mdelo 3 VAR(1)
P3<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 3,type ="none")
  if (max(roots(p1ct))<1)
  {P3[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
  else {P3[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P3==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P3 <- matrix(as.numeric(P3[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico3<-rowSums(P3)/(k-length(columnas_a_borrar))} else {Pronostico3<-rowSums(P3)/k}


# Mdelo 4 VAR(1) con constante
P4<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 1,type ="const")
  if (max(roots(p1ct))<1)
  {P4[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
  else {P4[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P4==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P4 <- matrix(as.numeric(P4[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico4<-rowSums(P4)/(k-length(columnas_a_borrar))} else {Pronostico4<-rowSums(P4)/k}

# Mdelo 5 VAR(2) con constante
  P5<-matrix(0, nrow = h, ncol =k )
  for (i in 1:k) {
    p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 2,type ="const")
    if (max(roots(p1ct))<1)
    {P5[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
    else {P5[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P5==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P5 <- matrix(as.numeric(P5[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
Pronostico5<-rowSums(P5)/(k-length(columnas_a_borrar))} else {Pronostico5<-rowSums(P5)/k}

# Mdelo 6 VAR(1) con tendencia
P6<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 1,type ="trend")
```

```
  if (max(roots(p1ct))<1)
  {P6[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
  else {P6[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P6==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P6 <- matrix(as.numeric(P6[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico6<-rowSums(P6)/(k-length(columnas_a_borrar))} else {Pronostico6<-rowSums(P6)/k}

# Mdelo 7 VAR(1) con tendencia y constante
P7<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  p1ct <- VAR(PIB[i:dim(PIB)[1],], p = 1,type ="trend")
  if (max(roots(p1ct))<1)
  {P7[,i]<-predict(p1ct,n.ahead = h)[[1]]$"PIB.edificaciones"[,1]}
  else {P7[,i]<-rep(0,h)}
}
porcentaje <- .6
columnas_a_borrar <- which(colMeans(P7==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P7 <- matrix(as.numeric(P7[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico7<-rowSums(P7)/(k-length(columnas_a_borrar))} else {Pronostico7<-rowSums(P7)/k}


#-----------------------Generar matriz de resultados--------------------------
Resutado1<-data.frame(Pronostico1,Pronostico2,Pronostico3,Pronostico4,Pronostico5,Pronostico6,Pronos

#-----------------------Selección de los modelos-------------------------------
z<-3

# Generar el criterio de seleccion
RMS<-(colSums((Resutado1-c(Prueba))*(Resutado1-c(Prueba))))/length(Prueba)
AMS<-(colSums(abs(Resutado1-c(Prueba))))/length(Prueba)
Combinado<-data.frame(Prome=(scale(RMS)+scale(AMS))/2)

orden_filas <- order(Combinado$Prome)
Modelos <- Combinado[orden_filas, ]
Modelos <- data.frame(row.names=c(rownames(Combinado)[orden_filas]),P=Modelos)
Modelos_selec_ARIMA<-rownames(Modelos)[1:z]

#--------------------------Construcción del pronostico conjunto ARIMA----------
#Promedio normal
Arima<-Resutado1[,Modelos_selec_ARIMA]
Arima$Promedio_Arima<-(rowSums(Arima))/3
Arima<-data.frame(Arima)

s <- as.Date("2023-03-01")
e <- as.Date("2023-12-01")
Fecha <- as.Date(seq(from = s, to = e, by = "3 months"))

Prueba<-data.frame(Prueba)
colnames(Prueba)<-"PIB"

Grafico <- data.frame(  "Modelo1" = Arima[, 1],  "Modelo2" = Arima[, 2],  "Modelo3" = Arima[, 3],  P
                        PIB=Prueba,
                        Fecha)
```

```
df <- melt(Grafico, id = c("Fecha"))

modelo_resaltado <- "PIB"

ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line(size = 0.8, linetype = "dashed") +  # Todas las líneas
  geom_line(data = subset(df, variable == modelo_resaltado), size = 1) +  # Línea resaltada
  xlab("Fecha") +
  ylab("Crecimiento PIB de edificaciones") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.005)) +
  labs(color = "Modelos")  # Etiqueta para la leyenda

#---------------Validación de las especificaciones VAR seleccionadas------------
# Modelo 1: Pronostico 5 modelo VAR con dos rezagos y constante.
# Modelo 2: Pronostico 2 modelo VAR con dos rezagos sin constante ni pendiente.
# Modelo 3: Pronostico 4 modelo VAR con un rezago sin constante ni pendiente.

rm(list=ls(all=TRUE))
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "B1:J77",sheet ="Base")),frequency = 4, start = c(2
PIB<-na.omit(window(PIB, start = c(2006, 1), end = c(2023, 4)))
colnames(PIB)[1]<-"Iniciada"
colnames(PIB)[2]<-"Culminada"
colnames(PIB)[3]<-"Licenciada"
colnames(PIB)[4]<-"r compra"
colnames(PIB)[5]<-"r constructor"
colnames(PIB)[6]<-"PIB edificaciones"
colnames(PIB)[7]<-"PIB"
colnames(PIB)[8]<-"Proceso"
colnames(PIB)[9]<-"Hogares"
gc()

#----- Validación modelo 1
Modelo_1 <- VAR(PIB, p = 2,type ="const")

# Estabilidad
stability <- stability(Modelo_1)
par(mar=c(2,2,2,2))
plot(stability)
# Correlación serial H_0 no correlación serial
serial <- serial.test(Modelo_1, lags.pt=16, type="PT.asymptotic")
print(serial)
# Prueba de heterocedasticidad H_0 no hay heteroceastividad
arch <- arch.test(Modelo_1, lags.multi=5)
print(arch)
# Prueba de normalidad de los residuos H_0 normalidad
normality <- normality.test(Modelo_1)
print(normality)
# Prueba de estacionariedad
residuals <- resid(Modelo_1)
kpss_test <- apply(residuals, 2, kpss.test)
kpss_test

#----- Validación modelo 2
Modelo_2 <- VAR(PIB, p = 2,type ="none")

# Estabilidad
stability <- stability(Modelo_2)
```

```
par(mar=c(2,2,2,2))
plot(stability)
# Correlación serial H_0 no correlación serial
serial <- serial.test(Modelo_2, lags.pt=16, type="PT.asymptotic")
print(serial)
# Prueba de heterocedasticidad H_0 no hay heteroceastividad
arch <- arch.test(Modelo_2, lags.multi=5)
print(arch)
# Prueba de normalidad de los residuos H_0 normalidad
normality <- normality.test(Modelo_2)
print(normality)
# Prueba de estacionariedad
residuals <- resid(Modelo_2)
kpss_test <- apply(residuals, 2, kpss.test)
kpss_test


#----- Validación modelo 3
Modelo_3 <- VAR(PIB, p = 1,type ="none")

# Estabilidad
stability <- stability(Modelo_3)
par(mar=c(2,2,2,2))
plot(stability)
# Correlación serial H_0 no correlación serial
serial <- serial.test(Modelo_3, lags.pt=16, type="PT.asymptotic")
print(serial)
# Prueba de heterocedasticidad H_0 no hay heteroceastividad
arch <- arch.test(Modelo_3, lags.multi=5)
print(arch)
# Prueba de normalidad de los residuos H_0 normalidad
normality <- normality.test(Modelo_3)
print(normality)
# Prueba de estacionariedad
residuals <- resid(Modelo_3)
kpss_test <- apply(residuals, 2, kpss.test)
kpss_test


#-------------------------------------------------------------------------------------------
#---------------------------------Probando modelos VARMA-----------------------------------
#-------------------------------------------------------------------------------------------
library(MTS)

rm(list=ls(all=TRUE))
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "B1:K77",sheet ="Base")),frequency = 4, start = c(2
PIB<-na.omit(window(PIB, start = c(2005, 2), end = c(2023, 4)))
colnames(PIB)[1]<-"Iniciada"
colnames(PIB)[2]<-"Culminada"
colnames(PIB)[3]<-"Licenciada"
colnames(PIB)[4]<-"r compra"
colnames(PIB)[5]<-"r constructor"
colnames(PIB)[6]<-"PIB edificaciones"
colnames(PIB)[7]<-"PIB"
colnames(PIB)[8]<-"Proceso"
colnames(PIB)[9]<-"Hogares"
colnames(PIB)[10]<-"Agregada"
```

```
# Estimación del modelo
model_1 <- VARMA(PIB[,c(6,10)], p = 1, q = 1)
ar_params <- model_1$coef[2,]

# Pronosticos manuales
last_values <- tail(PIB[,c(6,10)], 1)
h <- 10
forecasts <- matrix(NA, nrow = h, ncol = ncol(PIB[,c(6,10)]))

for (i in 1:h) {
  forecasts[i, ] <- (ar_params %*% last_values)[1,]
  last_values <- forecasts[i, ]
}

Pronostico_VARMA<-forecast

#-------------------------------------------------------------------------------------
#---------------------------------Modelos bayesianos----------------------------------
#-------------------------------------------------------------------------------------

#Se estiman modelos bayesianos tomando como enfoque los trabajos de Lutkepohl (2005, p. 225)
#se toma como función previa la dsitribucion Minnesota.
rm(list=ls(all=TRUE))
library(BVAR)

#Llamado de los datos
Tabla <- ts(read_excel("Base.xlsx",sheet = "Base",range = "b1:k77"),frequency = 4, start = c(2005, 1
tabla_1 <- na.omit(Tabla)
Tabla<-log(tabla_1)
Dif_tabla<-window(Tabla, start = c(2007, 3), end = c(2022, 4))
Prueba<-data.frame(window(Tabla,start=c(2023, 1),end=c(2023, 4)))[,"PIB_edificaciones"]
e1=Dif_tabla[,c(1,2,3,6,8)]

#Parametros iniciales
#Horizonte de tiempo para pronosticar
h<-length(Prueba)
#Maximo numero de datos a eliminar de la base en las estimaciones
k<-5

#-----------------Modelo1 minnesota un rezago
P2<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],], lags = 1,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P2[,i]<-(predict(x, horizon = h)$quants[, ,4 ])[2,]
}

Pronostico2<-rowSums(P2)/k


#-----------------Modelo2 minnesota dos rezagos
P3<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],], lags = 2,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P3[,i]<-(predict(x, horizon = h)$quants[, ,4 ])[2,]
}
```

```
Pronostico3<-rowSums(P3)/k


#-----------------Modelo3 minnesota tres rezagos
P4<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],], lags = 3,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P4[,i]<-(predict(x, horizon = h)$quants[, ,4 ])[2,]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P4==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P4 <- matrix(as.numeric(P4[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico4<-rowSums(P4)/(k-length(columnas_a_borrar))} else {Pronostico4<-rowSums(P4)/k}

#-----------------Modelo4 minnesota cuatro rezagos
P5<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],], lags = 4,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P5[,i]<-(predict(x, horizon = h)$quants[, ,4 ])[2,]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P5==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P5 <- matrix(as.numeric(P5[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico5<-rowSums(P5)/(k-length(columnas_a_borrar))} else {Pronostico5<-rowSums(P5)/k}

#-----------------Modelo5 minnesota un rezago PIB y y sin licencias
P6<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],-3], lags = 1,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P6[,i]<-(predict(x, horizon = h)$quants[, ,3 ])[2,]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P6==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P6 <- matrix(as.numeric(P6[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico6<-rowSums(P6)/(k-length(columnas_a_borrar))} else {Pronostico6<-rowSums(P6)/k}

#-----------------Modelo6 minnesota dos rezagos PIB y sin licencias
P7<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],-3], lags = 2,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P7[,i]<-(predict(x, horizon = h)$quants[, ,3 ])[2,]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P7==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P7 <- matrix(as.numeric(P7[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
```

```
  Pronostico7<-rowSums(P7)/(k-length(columnas_a_borrar))} else {Pronostico7<-rowSums(P7)/k}

#-----------------Modelo6 minnesota tres rezagos PIB y sin licencias
P8<-matrix(0, nrow = h, ncol =k )
for (i in 1:k) {
  x <-BVAR::bvar(data = e1[i:dim(e1)[1],-3], lags = 3,bv_priors="minnesota",
                 n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
  P8[,i]<-(predict(x, horizon = h)$quants[, ,3 ])[2,]
}

porcentaje <- .6
columnas_a_borrar <- which(colMeans(P8==0)>porcentaje)
if (length(columnas_a_borrar)>0){
  P8 <- matrix(as.numeric(P8[,-columnas_a_borrar ]),nrow=h,ncol=k-length(columnas_a_borrar ))
  Pronostico8<-rowSums(P8)/(k-length(columnas_a_borrar))} else {Pronostico8<-rowSums(P8)/k}


#-----------------------Generar matriz de resultados--------------------------
Resutado1<-data.frame(Pronostico2,Pronostico3,Pronostico4,Pronostico5,Pronostico6,Pronostico7,Pronos

#-----------------------Selección de los modelos-------------------------------

#-----------------------------------Selección de los modelos--------------------
z<-3
RMS<-(colSums((Resutado1-c(Prueba))*(Resutado1-c(Prueba))))/length(Prueba)
AMS<-(colSums(abs(Resutado1-c(Prueba))))/length(Prueba)
Promedio<-data.frame(Prome=(scale(RMS)+scale(AMS))/2)

orden_filas <- order(Promedio$Prome)
Modelos <- Promedio[orden_filas, ]
Modelos <- data.frame(row.names=c(rownames(Promedio)[orden_filas]),Promedio=Modelos)
Modelos_selec_ARIMA<-rownames(Modelos)[1:z]



#--------------------------Construcción del pronostico conjunto ARIMA----------
#Promedio normal
Arima<-Resutado1[,Modelos_selec_ARIMA]
Arima$Promedio_Arima<-(rowSums(Arima))/3


s <- as.Date("2023-03-01")
e <- as.Date("2023-12-01")
Fecha <- as.Date(seq(from = s, to = e, by = "3 months"))

Prueba<-data.frame(Prueba)
colnames(Prueba)<-"PIB"
Grafico <- data.frame(  "Modelo1" = Arima[, 1],  "Modelo2" = Arima[, 2],  "Modelo3" = Arima[, 3],  P
                        PIB=Prueba,
                        Fecha)

df <- melt(Grafico, id = c("Fecha"))

modelo_resaltado <- "PIB"

ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line(size = 0.8, linetype = "dashed") +  # Todas las líneas
  geom_line(data = subset(df, variable == modelo_resaltado), size = 1) +  # Línea resaltada
```

```
  xlab("Fecha") +
  ylab("Logaritmo del PIB de edificaciones") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.005)) +
  labs(color = "Modelos")  # Etiqueta para la leyenda



#-------------------------------------------------------------------------------------------------------
#----Proyección final para 2024, 2025 y 2026 con base en los modelos seleccionados en esta sección d
#-------------------------------------------------------------------------------------------------------

#------------------
#---- Modelos ARIMA
#------------------
rm(list=ls(all=TRUE))
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "G1:G77",sheet ="Base")),frequency = 4, start = c(2
PIB[58]=(c(-0.01587182))

# Estimación de los modelos elegidos
Modelo_1<-arima(PIB, order=c(3, 0, 1))
Modelo_2<-arima(PIB, order=c(2, 0, 4))
Modelo_3<-arima(PIB, order=c(3, 0, 2))

# Estimación de las sendas para 2024 y 2025

Pronostico01<-c(predict(Modelo_1,8)$pred)
Pronostico02<-c(predict(Modelo_2,8)$pred)
Pronostico03<-c(predict(Modelo_3,8)$pred)


#----------------
#---- Modelos VAR
#----------------

#---------------Validación de las especificaciones VAR seleccionadas------------
# Modelo 1: Pronostico 5 modelo VAR con dos rezagos y constante.
# Modelo 2: Pronostico 2 modelo VAR con dos rezagos sin constante ni pendiente.
# Modelo 3: Pronostico 4 modelo VAR con un rezago sin constante ni pendiente.

library(vars)
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "B1:J77",sheet ="Base")),frequency = 4, start = c(2
PIB<-na.omit(window(PIB, start = c(2006, 1), end = c(2023, 4)))
colnames(PIB)[1]<-"Iniciada"
colnames(PIB)[2]<-"Culminada"
colnames(PIB)[3]<-"Licenciada"
colnames(PIB)[4]<-"r compra"
colnames(PIB)[5]<-"r constructor"
colnames(PIB)[6]<-"PIB edificaciones"
colnames(PIB)[7]<-"PIB"
colnames(PIB)[8]<-"Proceso"
colnames(PIB)[9]<-"Hogares"


Modelo_4 <- vars::VAR(PIB, p = 2,type ="const")
Pronostico04<-predict(Modelo_4,n.ahead = 8)[[1]]$"PIB.edificaciones"[,1]
```

```
Modelo_5 <- vars::VAR(PIB, p = 2,type ="none")
Pronostico05<-predict(Modelo_5,n.ahead = 8)[[1]]$"PIB.edificaciones"[,1]

Modelo_6 <- vars::VAR(PIB, p = 1,type ="none")
Pronostico06<-predict(Modelo_6,n.ahead = 8)[[1]]$"PIB.edificaciones"[,1]

#------------
#Modelo VARMA
#------------
setwd("D:/Articulo redes/Bases y programaciones/")
PIB<- diff(ts(log(read_excel("Base.xlsx",range = "B1:K77",sheet ="Base")),frequency = 4, start = c(2
PIB<-na.omit(window(PIB, start = c(2005, 2), end = c(2023, 4)))
colnames(PIB)[1]<-"Iniciada"
colnames(PIB)[2]<-"Culminada"
colnames(PIB)[3]<-"Licenciada"
colnames(PIB)[4]<-"r compra"
colnames(PIB)[5]<-"r constructor"
colnames(PIB)[6]<-"PIB edificaciones"
colnames(PIB)[7]<-"PIB"
colnames(PIB)[8]<-"Proceso"
colnames(PIB)[9]<-"Hogares"
colnames(PIB)[10]<-"Agregada"

# Estimación del modelo
library(MTS)
model_1 <- VARMA(PIB[,c(6,10)], p = 1, q = 1)
ar_params <- model_1$coef[2,]

# Pronosticos manuales
last_values <- tail(PIB[,c(6,10)], 1)
h <- 8
forecasts <- matrix(NA, nrow = h, ncol = ncol(PIB[,c(6,10)]))

for (i in 1:h) {
  forecasts[i, ] <- (ar_params %*% last_values)[1,]
  last_values <- forecasts[i, ]
}

Pronostico_VARMA<-forecasts[,1]


#------------------
#Modelos bayesianos
#-----------------

Tabla <- ts(read_excel("Base.xlsx",sheet = "Base",range = "b1:k77"),frequency = 4, start = c(2005, 1
tabla_1 <- na.omit(Tabla)
Tabla<-log(tabla_1)
Dif_tabla<-window(Tabla, start = c(2007, 3), end = c(2023, 4))
e1=Dif_tabla[,c(1,2,3,6,8)]
h=8
x <-BVAR::bvar(data = e1, lags = 4,bv_priors="minnesota",
               n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
Pronostico09<-(predict(x, horizon = h)$quants[, ,4 ])[2,]

x <-BVAR::bvar(data = e1, lags = 3,bv_priors="minnesota",
               n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
Pronostico10<-(predict(x, horizon = h)$quants[, ,4 ])[2,]
```

```
x <-BVAR::bvar(data = e1[,-3], lags = 3,bv_priors="minnesota",
               n_draw = 15000, n_burn = 400, n_thin = 2, verbose = FALSE)
Pronostico11<-(predict(x, horizon = h)$quants[, ,3 ])[2,]



Resutado1<-data.frame(Pronostico01,Pronostico02,Pronostico03,Pronostico04,
                      Pronostico05,Pronostico06,Pronostico_VARMA,Pronostico09,Pronostico10,Pronostic


write.xlsx(Resutado1, "Resutado_clasicos_finales.xlsx", dec = ",", rowNames = FALSE)




#-------------------------------------------------------------------------------------------------------
#-------------------------------------------------------------------------------------------------------
#-------------------------------------------------------------------------------------------------------
#---------------------------------------Progrmaci?n redes neuronales-----------------------------------

#-------------------------------------------------------------------------------------------------------
#--- Descripcion: esta programacion realiza la estimacion de los pronosticos del PIB de edificacione
# utilizando la metodolog?a de redes neuronales.
#--- Autores: Daniela Sedano y Cristian Torres
#--- Actualizacion: 01-08-2024

#-----------------------------------------------Paquetes necesarios-----------------------------------

## Instalaci?n, solo es necesario correrce una vez
install.packages("neuralnet")
install.packages("caret")
#install.packages("vctrs")

##Llamado de las librerias
library(neuralnet)
library(stats)
library(readxl)
library(dplyr)
library(caret)

#-----------------------------------------------Llamado de los datos----------------------------------
## Cambiar directorio, se debe cambiar por la ruta donde se encuentren la base a ser utilizada
setwd("D:/Articulo redes/Bases y programaciones/")
rm(list=ls(all=TRUE))
set.seed(123)
## Importacion de la tabla
Base <-read_excel("Base.xlsx",range = "B1:B77",sheet ="PIB")
##Los datos se convierten en serie de tiempo
PIB<- na.omit(diff(ts(log(Base),frequency = 4, start = c(2005, 1)),4))

## Se transforma en un data.frame para facilitar el uso de la funci?n "lag".
final<-data.frame(PIB)
colnames(final)[1]<-"Edificaciones"

## Crear la base con rezagos
datos_lag <- cbind(lag(final$Edificaciones, 1), lag(final$Edificaciones, 2), lag(final$Edificaciones
```

```
## Convertir de serie temporal la base lag a data.frame
datos_df <- as.data.frame(datos_lag)

## Cambiar nombres de columnas
colnames(datos_df) <- c("lag1", "lag2", "lag3")
dim(datos_df)

##Remplazar los vacios generados por los rezagos con la media de cada trimestre
Trimestre<-c(rep(c("I","II","III","IV"),18))
final<-data.frame(Edificaciones=PIB,Trimestre)
Media<-final%>%group_by(Trimestre)%>%
  summarise_all(mean)

a<-Media$PIB_edificaciones[1]
b<-Media$PIB_edificaciones[2]
c<-Media$PIB_edificaciones[3]
d<-Media$PIB_edificaciones[4]

datos<- cbind(PIB=final[,-2],datos_df)
datos$lag1[1] <-d
datos$lag2[1]<-d
datos$lag3[1]<-d
datos$lag2[2]<-a
datos$lag3[2]<-a
datos$lag3[3]<-b

#-------------------------------------------------------------------------------
#---------------En la primera parte probamos las especificaciones--------------
#--------------dentro de la muestra "tomamos los trimestresde 2023-------------
#---------------------Para seleccionar las especificaciones---------------------
#-------------------------------------------------------------------------------

#------Modelo 1 tres rezagos dos capas ocultas con 1 neurona por capa oculta

nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(1,1))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_1<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 2 tres rezagos dos capas ocultas con 2 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(2,2))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
```

```
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_2<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 3 tres rezagos dos capas ocultas con 3 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(3,3))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_3<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 4 tres rezagos dos capas ocultas con 4 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(4,4))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_4<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 5 tres rezagos dos capas ocultas con 5 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(5,5))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
```

```
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_5<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)


#------Modelo 6 tres rezagos dos capas ocultas con 6 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(6,6))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_6<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 7 tres rezagos dos capas ocultas con 7 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(7,7))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_7<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 8 tres rezagos dos capas ocultas con 8 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(8,8))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
```

```
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_8<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)


#------Modelo 9 tres rezagos dos capas ocultas con 9 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(9,9))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_9<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 10 tres rezagos dos capas ocultas con 10 neuronas por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(10,10))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_10<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 11 tres rezagos tres capas ocultas con 1 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(1,1,1))
testeo<-datos[c(69:72),]
plot(nn)

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
```

```
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_11<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)


#------Modelo 12 tres rezagos tres capas ocultas con 2 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(2,2,2))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_12<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 13 tres rezagos tres capas ocultas con 3 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(3,3,3))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_13<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 14 tres rezagos tres capas ocultas con 4 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(4,4,4))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
```

```
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_14<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 15 tres rezagos tres capas ocultas con 5 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(5,5,5))
testeo<-datos[c(69:72),]
plot(nn)

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_15<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 16 tres rezagos tres capas ocultas con 6 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(6,6,6))
testeo<-datos[c(69:72),]

#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_16<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)


#------Modelo 17 tres rezagos tres capas ocultas con 7 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(7,7,7))
testeo<-datos[c(69:72),]

#Paso1
```

```
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_17<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 18 tres rezagos tres capas ocultas con 8 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(8,8,8))
testeo<-datos[c(69:72),]


#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_18<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 19 tres rezagos tres capas ocultas con 9 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(9,9,9))
testeo<-datos[c(69:72),]


#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_19<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

#------Modelo 20 tres rezagos tres capas ocultas con 10 neurona por capa oculta
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos[c(-69:-72),],hidden = c(10,10,10))
testeo<-datos[c(69:72),]
```

```
#Paso1
Prueba<-t(c(-0.084567642,0.241099480,0.034236288))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,-0.084567642,0.241099480))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,-0.084567642))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)

Pronostico_20<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)


#-----------------------Generar matriz de resultados--------------------------
Resutado1<-data.frame(Pronostico_1,Pronostico_2,Pronostico_3,Pronostico_4,
                      Pronostico_5,Pronostico_6,Pronostico_7
                      ,Pronostico_8,Pronostico_9,Pronostico_10,Pronostico_11,
                      Pronostico_12,Pronostico_13,Pronostico_14,Pronostico_15,Pronostico_16
                      ,Pronostico_17,Pronostico_18,Pronostico_19,Pronostico_20)

#-----------------------Selección de los modelos-------------------------------
z<-3
Prueba<-testeo[,1]
# Generar el criterio de seleccion
RMS<-(colSums((Resutado1-c(Prueba))*(Resutado1-c(Prueba))))/length(Prueba)
AMS<-(colSums(abs(Resutado1-c(Prueba))))/length(Prueba)
Combinado<-data.frame(Prome=(scale(RMS)+scale(AMS))/2)

orden_filas <- order(Combinado$Prome)
Modelos <- Combinado[orden_filas, ]
Modelos <- data.frame(row.names=c(rownames(Combinado)[orden_filas]),P=Modelos)
Modelos_selec_ARIMA<-rownames(Modelos)[1:z]

#---------------------------Construcción del pronostico conjunto ARIMA----------
#Promedio normal
Arima<-Resutado1[,Modelos_selec_ARIMA]
Arima$Promedio_Arima<-(rowSums(Arima))/3
Arima<-data.frame(Arima)

s <- as.Date("2023-03-01")
e <- as.Date("2023-12-01")
Fecha <- as.Date(seq(from = s, to = e, by = "3 months"))

Prueba<-data.frame(Prueba)
colnames(Prueba)<-"PIB"

Grafico <- data.frame(  "Modelo1" = Arima[, 1],  "Modelo2" = Arima[, 2],  "Modelo3" = Arima[, 3],  P
                        PIB=Prueba,
                        Fecha)
library("reshape2")
df <- melt(Grafico, id = c("Fecha"))

modelo_resaltado <- "PIB"
```

```
ggplot(df, aes(x = Fecha, y = value, col = variable)) +
  geom_line(size = 0.8, linetype = "dashed") +  # Todas las líneas
  geom_line(data = subset(df, variable == modelo_resaltado), size = 1) +  # Línea resaltada
  xlab("Fecha") +
  ylab("Crecimiento PIB de edificaciones") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.005)) +
  labs(color = "Modelos")  # Etiqueta para la leyenda




#-------------------------------------------------------------------------------
#---------------Se tomas las especificaciones con mejores resultados -----------
#-----------------------y se proyecta 2024 y 2025--------------------------------
#-------------------------------------------------------------------------------
set.seed(123)
#-----------------------------------------------------------------(1)


nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos,hidden = c(5,5))
plot(nn)
#Paso1
Prueba<-t(c(0.044183078,-0.065967400,0.021795697))
predicciones_1 <- predict(nn,newdata=Prueba)
#Paso2
Prueba<-t(c(predicciones_1,0.044183078,-0.065967400))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,0.044183078))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)
#Paso5
Prueba<-t(c(predicciones_4,predicciones_3,predicciones_2))
predicciones_5 <- predict(nn,newdata=Prueba)
#Paso6
Prueba<-t(c(predicciones_5,predicciones_4,predicciones_3))
predicciones_6 <- predict(nn,newdata=Prueba)
#Paso7
Prueba<-t(c(predicciones_6,predicciones_5,predicciones_4))
predicciones_7 <- predict(nn,newdata=Prueba)
#Paso8
Prueba<-t(c(predicciones_7,predicciones_6,predicciones_5))
predicciones_8 <- predict(nn,newdata=Prueba)

Pronostico_1<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4,predicciones_5,
                predicciones_6,predicciones_7,predicciones_8)

plot(ts(Pronostico_1))

#-----------------------------------------------------------------(2)
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos,hidden = c(8,8))
plot(nn)
#Paso1
Prueba<-t(c(0.044183078,-0.065967400,0.021795697))
predicciones_1 <- predict(nn,newdata=Prueba)
```

```
#Paso2
Prueba<-t(c(predicciones_1,0.044183078,-0.065967400))
predicciones_2 <- predict(nn,newdata=Prueba)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,0.044183078))
predicciones_3 <- predict(nn,newdata=Prueba)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba)
#Paso5
Prueba<-t(c(predicciones_4,predicciones_3,predicciones_2))
predicciones_5 <- predict(nn,newdata=Prueba)
#Paso6
Prueba<-t(c(predicciones_5,predicciones_4,predicciones_3))
predicciones_6 <- predict(nn,newdata=Prueba)
#Paso7
Prueba<-t(c(predicciones_6,predicciones_5,predicciones_4))
predicciones_7 <- predict(nn,newdata=Prueba)
#Paso8
Prueba<-t(c(predicciones_7,predicciones_6,predicciones_5))
predicciones_8 <- predict(nn,newdata=Prueba)

Pronostico_2<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4,predicciones_5,
                predicciones_6,predicciones_7,predicciones_8)

plot(ts(Pronostico_2))

#-----------------------------------------------------------------(3)
nn<-neuralnet(PIB~lag1+lag2+lag3,data = datos,hidden = c(1,1))
plot(nn)
#Paso1
Prueba<-t(c(0.248379008,0.048763842,0.127703220))
predicciones_1 <- predict(nn,newdata=Prueba,n.ahead = 10)
#Paso2
Prueba<-t(c(predicciones_1,0.248379008,0.048763842))
predicciones_2 <- predict(nn,newdata=Prueba,n.ahead = 10)
#Paso3
Prueba<-t(c(predicciones_2,predicciones_1,0.248379008))
predicciones_3 <- predict(nn,newdata=Prueba,n.ahead = 10)
#Paso4
Prueba<-t(c(predicciones_3,predicciones_2,predicciones_1))
predicciones_4 <- predict(nn,newdata=Prueba,n.ahead = 10)
Pronostico_3<-c(predicciones_1,predicciones_2,predicciones_3,predicciones_4)

library(openxlsx)
Resultado_RNA<-data.frame(Pronostico_1,Pronostico_2,Pronostico_3)
write.xlsx(Resultado_RNA, "Resultado_RNA.xlsx", dec = ",", rowNames = FALSE)
```