



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ciencias Naturales y Matemáticas

AUTOMATIZACIÓN DEL PRONOSTICO MULTIVARIANTE DE ENFERMEDADES

FÚNGICAS DEL CACAO UTILIZANDO R

MATE-219

PROYECTO INTEGRADOR

Previo a la obtención del Título de:

Matemático

Presentado por:

Elías Abraham Jumbo Rodriguez

Diego Fernando López Campoverde

GUAYAQUIL - ECUADOR

Año: 2025

DEDICATORIA

Dedico el presente proyecto a mi familia, especialmente a mis padres, quienes me han apoyado en cada decisión que he tomado, que han sabido guiarme y aconsejarme. Sobre todo, que con su amor siempre me motivaron a ser mejor y a superarme cada día.

Elías Jumbo R.

DEDICATORIA

A mi hijo Demian y a mi esposa Alison,
quienes han sido el motivo para seguir
adelante y superarme cada día.

Diego López C.

AGRADECIMIENTOS

Agradezco a mis padres, a mis hermanos y mis seres queridos, por su apoyo, sus consejos y su amor. A mis maestros, quienes fueron guía e inspiración, que me enseñaron el valor del esfuerzo y la perseverancia. A mis amigos, quienes se mantuvieron conmigo durante todo mi camino, con apoyo y buenos momentos. Quiero agradecer especialmente a Indira Sornoza, quien con su apoyo, amor y confianza en mí, me llenó valor y fuerzas para continuar avanzando y superarme cada día. A nuestro Tutor, Omar Ruíz, que nos ha guiado durante todo el desarrollo del proyecto.

Elías Jumbo R.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a Omar Ruiz, por su guía y paciencia durante todo el desarrollo de este proyecto. Agradezco también a la profesora Elimar Marchan, por su apoyo, motivación y dedicación en cada etapa del trabajo. Asimismo, quiero reconocer a Elías Jumbo, por su colaboración, compromiso y compañerismo, que hicieron más llevadero este proceso.

Diego López C.

Declaración Expresa

Nosotros Diego Fernando López Campoverde y Elías Abraham Jumbo Rodríguez acordamos acuerdo/acordamos y reconozco/reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

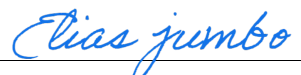
La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al/los autor/es que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 28 de mayo de 2025



Diego López C.



Elías Jumbo R.

EVALUADORES

Luz Elimar Marchan, Ph.D.

PROFESORA DE LA MATERIA

Omar Ruíz, Ph.D

TUTOR

RESUMEN

El cultivo de cacao en Ecuador enfrenta pérdidas significativas por enfermedades fúngicas como moniliasis (*Moniliophthora roreri*) y fitóftora (*Phytophthora palmivora*), favorecidas por condiciones climáticas húmedas. La ausencia de pronósticos precisos limita la capacidad de los agricultores para anticipar brotes y aplicar medidas preventivas de manera oportuna. El propósito de este proyecto es proporcionar una herramienta en R que automatice la construcción y evaluación de modelos de pronóstico multivariantes, integrando covariables climáticas y sanitarias, con el fin de mejorar la toma de decisiones agrícolas. Se consideran modelos ARIMA, SARIMA, ARIMAX, SARIMAX, VARMA, VARMAX, ETS y Prophet, aplicados en R Shiny, utilizando limpieza de datos, imputación y validación cruzada con ventanas móviles, evaluando RMSE, MAPE, AIC y Theil's U. La inclusión de covariables como la precipitación permite optimizar la precisión frente a modelos univariados y generar pronósticos más robustos para la gestión preventiva de enfermedades fúngicas.

Palabras clave: cacao, enfermedades fúngicas, series temporales, R Shiny.

ABSTRACT

*Cocoa cultivation in Ecuador faces significant losses due to fungal diseases such as frosty pod rot (*Moniliophthora roreri*) and black pod disease (*Phytophthora palmivora*), which are promoted by humid climatic conditions. The lack of accurate forecasts limits farmers' ability to anticipate outbreaks and implement timely preventive measures. The purpose of this project is to provide an R-based tool that automates the construction and evaluation of multivariate forecasting models, integrating climatic and sanitary covariates to improve agricultural decision-making. ARIMA, SARIMA, ARIMAX, SARIMAX, VARMA, VARMAX, ETS, and Prophet models are considered and applied in R Shiny, using data cleaning, imputation, and rolling-window cross-validation, and evaluated with RMSE, MAPE, AIC, and Theil's U. Including covariates such as precipitation enhances forecast accuracy compared to univariate models and generates more robust predictions for the preventive management of fungal diseases.*

Keywords: *cocoa, fungal diseases, time series models, R Shiny.*

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ABREVIATURAS	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. INTRODUCCIÓN	1
1.1 Descripción del problema	2
1.2 Justificación del problema	3
1.3 Objetivos	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	5
1.4 Marco teórico	5
1.4.1 Moniliasis y Phitophtora	5
1.4.2 Impácto económico	6
1.4.3 Modelos de series temporales	8
1.4.4 Evaluación de Modelos	13
1.4.5 Prueba de Dickey-Fuller Aumentada (ADF)	14
CAPÍTULO 2	16

2. METODOLOGÍA	16
2.1 Formulacion de alternativas	16
2.2 Diseño del modelo utilizado	17
2.3 Preparacion de datos	18
2.4 Especificaciones técnicas	19
2.5 Robustez y escalabilidad	21
2.6 Reproductibilidad y despliegue	21
2.7 Consideraciones éticas y legales	21
CAPÍTULO 3	23
3. RESULTADOS Y ANÁLISIS	23
3.1 Descripción de los datos y preprocesamiento	23
3.2 Estacionariedad y estructura de dependencia	24
3.3 Correlación entre variables	25
3.3.1 Variables numéricas utilizadas	25
3.4 Modelado y validación	26
3.5 Comparativo de desempeño	27
3.6 Pronósticos	28
3.7 Estimacion de ganancias económicas mensuales	32
3.8 Discusión de resultados	32
CAPÍTULO 4	34
4. CONCLUSIONES Y RECOMENDACIONES	34
BIBLIOGRAFÍA	

APÉNDICES

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
SARMA	Seasonal Auto Regressive Moving Average
SARIMA	Seasonal Auto Regressive Integrated Moving Average
ARIMAX	Auto Regressive Integrated Moving Average with Exogenous variables
SARIMAX	Seasonal Auto Regressive Integrated Moving Average with Exogenous variables
VARMA	Vector Auto Regressive Moving Average
VARMAX	Vector Auto Regressive Moving Average with Exogenous variables
ETS	Error, Trend, Seasonal (Exponential Smoothing)
Prophet	Forecasting model developed by Facebook (Additive or Multiplicative Time Series Model)

ÍNDICE DE FIGURAS

Figura 3.1	Evolutivo de producción del cacao desde el año 2022 al 2024 según meses. . .	24
Figura 3.2	Pronóstico del mejor modelo (ARIMAX).	28
Figura 3.3	Pronóstico del modelo <i>ETS</i>	29
Figura 3.4	Pronóstico del modelo <i>ARIMA</i>	29
Figura 3.5	Pronóstico del modelo <i>SARIMA</i>	30
Figura 3.6	Pronóstico del modelo <i>PROPHET</i>	30
Figura 3.7	Pronóstico del modelo <i>VAR</i>	31
Figura 3.8	Comparación de pronósticos: <i>ETS</i> , <i>ARIMA</i> , <i>ARIMAX</i> , <i>SARIMAX</i> , <i>PROPHET</i> , <i>SARIMA</i> , <i>VAR</i>	31

ÍNDICE DE TABLAS

Tabla 1.1	Pérdidas anuales estimadas en la producción de cacao en Ecuador (2020–2023).	7
Tabla 3.1	Estadísticas descriptivas de la producción mensual de cacao (toneladas).	24
Tabla 3.2	Matriz de correlación entre variables numéricas.	26
Tabla 3.3	Comparativo de modelos (métricas promedio).	28
Tabla 3.4	Producción proyectada y valor económico mensual (ARIMAX, 2025)	32

CAPÍTULO 1

1. INTRODUCCIÓN

El (*Theobroma cacao*) constituye uno de los principales productos agrícolas de exportación en Ecuador, no solo por su importancia económica sino también por su valor cultural e histórico. Su producción enfrenta constantes amenazas, debido a enfermedades fúngicas tales como la moniliasis (*Moniliophthora roreri*) y la fitóftora (*Phytophthora palmivora*), las cuales afectan directamente la calidad y la cantidad de la cosecha.

Estas enfermedades se propagan con gran facilidad en ambientes con alta humedad y poco ventilados, generando así pérdidas en las cosechas en determinadas regiones.

El impacto de estas enfermedades se ve agravado por una inadecuada gestión agrícola y también por la falta de herramientas analíticas para poder anticiparse a los brotes. La detección temprana y el control oportuno de estas enfermedades requieren el uso de modelos predictivos capaces de integrar múltiples variables, como lo son la temperatura, la humedad, la precipitación, la incidencia de monilia y fitóftora.

Sin embargo, muchos de los enfoques en la actualidad carecen de una visión multivariada o no aprovechan eficazmente la información que se tiene disponible.

Debido a esta problemática, el presente proyecto tiene como objetivo el desarrollo de una función que automatice la construcción de modelos de predicción multivariantes para enfermedades fúngicas en el cultivo de cacao. Esta herramienta permitirá integrar covariables

climáticas y sanitarias en modelos de series temporales como lo son ARIMA, SARIMA, ARIMAX, SARIMAX, VARMA, VARMAX y Prophet, mejorando así la precisión de los pronósticos y facilitando la toma de decisiones informadas en el campo.

1.1 Descripción del problema

La moniliasis es una enfermedad fúngica causada por diferentes especies del género *Monilinia*, en particular, *Monilinia fructicola* y *Monilinia laxa*. Aunque estas especies se identifican inicialmente en frutas de climas templados, en el cacao también se han observado síntomas similares ocasionados por patógenos como (*moniliophthora rorei*), que anteriormente se clasificaba como parte del género *Monilia*. Esta clase de hongo se dirige principalmente a los frutos verdes del cacao, generando daños que resultan en su deterioro anticipado. La propagación del hongo ocurre más fácilmente en ambientes con alta humedad y poca circulación de aire.

La moniliasis ha tenido efectos catastróficos en países vecinos como en México tras la introducción de esta plaga en el 2005, la producción cacaotera disminuyó aproximadamente 60% y en un año cálido como 2013 llegó a afectar hasta el 70% de la cosecha nacional, según García (2015). En regiones productoras como Chiapas y Tabasco, las pérdidas alcanzaron la mitad de la producción anual (García, 2015).

En Colombia, se estima que la moniliasis causa la pérdida de alrededor del 40% de la producción de cacao seco cada año (Rodríguez and Pérez, 2019). De igual manera, en Nicaragua se ha documentado que la enfermedad puede devastar por completo plantaciones, con hasta un 100% de pérdida en los frutos (Martínez et al., 2017).

Estos datos ilustran la enorme brecha entre la producción potencial de cacao y la producción real

obtenida bajo la presencia de la *Moniliophthora roreri*. La moniliasis conlleva pérdidas directas de toneladas de cacao que se traducen en millones de dólares no percibidos, amenazando la sostenibilidad económica de la cultura cacaotera en Ecuador y la región.

Por otra parte, la fitóftora, causante de la pudrición parda en el cacao, es considerada una de las plagas más destructivas para este cultivo. En Ecuador, se ha detectado la existencia de fitóftora palmivora, que afecta tanto los frutos como los tallos y raíces de los árboles. Esta enfermedad se propaga en entornos con una humedad alta y constante, ya sea debido a lluvias continuas o a un manejo ineficaz del riego. La infección por fitóftora puede disminuir la producción anual entre un 30% y un 40% en plantaciones sin tratamiento, de acuerdo con INIAP (2020).

La influencia de estas patologías se intensifica por elementos vinculados a la administración del cultivo. Esto incluye técnicas agrícolas inadecuadas, como la poda errónea, la falta de supervisión constante y una gestión fitosanitaria deficiente. Asimismo, las condiciones cambiantes del entorno, como la temperatura y la humedad, complican la aplicación de estrategias de prevención consistentes.

El manejo inadecuado de variables en los modelos de predicción empeora la situación, dificultando la identificación temprana de brotes y la distribución óptima de recursos. Dando como resultado que la elección de ciertos modelos obstaculice una interpretación adecuada de la dinámica de estas enfermedades dentro del contexto tratado.

1.2 Justificación del problema

El proyecto nace de la necesidad de encontrar una solución práctica y efectiva para uno de los principales problemas que enfrentan los productores de cacao en Ecuador: las

enfermedades fúngicas como la moniliasis y la fitóftora. Estas enfermedades afectan directamente a la producción, causando pérdidas económicas importantes, especialmente en zonas con climas húmedos donde estos hongos se propagan con facilidad.

La idea de desarrollar una aplicación estadística que ayude a predecir la aparición de estas enfermedades surge porque actualmente muchos de los métodos que se utilizan no toman en cuenta todas las variables que influyen en el problema, como el clima, el tipo de manejo agrícola o la frecuencia de los brotes. Además, muchas veces los datos disponibles no se aprovechan bien por falta de herramientas adecuadas para analizarlos.

Este proyecto es importante porque puede ayudar a mejorar el control de las enfermedades y a planificar mejor las acciones de manejo en el campo. A través del uso de modelos de series temporales con covariables, se pueden generar pronósticos más precisos, lo que permite actuar a tiempo y reducir el impacto de los brotes. Aquí es donde interviene la matemática, ya que los modelos que se aplican requieren conocimientos en estadística, probabilidad y análisis de datos. La solución que se propone beneficiaría directamente a los técnicos y productores agrícolas, pero también a instituciones de investigación y al sector académico. Además, al tratarse de un desarrollo en R, puede adaptarse y mejorar con el tiempo, lo que abre la puerta a que otros investigadores y estudiantes lo usen como base para seguir innovando.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar una función en R, que automatice la generación de pronósticos multivariantes de enfermedades fúngicas en el cultivo de cacao integrando covariables, ofreciendo alertas

tempranas para el control de plagas.

1.3.2 Objetivos Específicos

- Explorar series de datos temporales que incluyan variables de incidencia de enfermedades (como moniliasis y fitóftora), junto con covariables climáticas;
- Desarrollar una función en R que permita al usuario el uso de su base de datos y la obtención automatizada de modelos adecuados, con el diagnóstico de supuestos y pronósticos para múltiples variables;
- Validar la herramienta desarrollada con un conjuntos de datos simulados, documentando su uso con ejemplos reproducibles.

1.4 Marco teórico

1.4.1 Moniliasis y *Phitophtora*

El cultivo de cacao en Ecuador representa uno de los pilares de la economía agrícola del país, destacándose como uno de los principales productos de exportación.

Sin embargo, la productividad de este cultivo está sujeta a una serie de factores exógenos, entre los cuales destacan las condiciones climáticas como la temperatura, precipitación, radiación solar y las enfermedades fúngicas, en especial, la moniliasis (*Moniliophthora roreri*) y la fitóftora (*Phytophthora spp*). Según el Instituto Nacional de Investigación Agropecuaria INIAP (2020), estas enfermedades pueden provocar hasta un 40% de pérdidas en la producción anual en zonas de alta incidencia.

Diversos estudios han demostrado la efectividad de los modelos estadísticos para

pronosticar la producción de cacao en diferentes contextos. Pinto et al. (2015) aplicaron modelos ARIMA para predecir la producción de cacao en Brasil, encontrando correlaciones significativas con variables climáticas como la temperatura y la humedad relativa.

En Ecuador, Muñoz et al. (2022) emplearon modelos SARIMA y ARIMAX para analizar la producción mensual de cacao y concluyeron que agregar variables climáticas como factores predictivos mejoraba notablemente la precisión del modelo. Asimismo, Galarza et al. (2021) exploraron el uso de modelos multivariantes (VAR) para capturar las interdependencias entre el clima, la salud del cultivo y los volúmenes de exportación de cacao ecuatoriano.

1.4.2 *Impacto económico*

Diversos reportes del Ministerio de Agricultura y Ganadería y la FAO destacan que la moniliasis puede causar pérdidas de entre el 30% y el 40% de la producción anual, mientras que la fitóftora provoca un impacto estimado de entre 10% y 20% MAG (2021),FAO (2025). La coexistencia de ambas enfermedades ha elevado los niveles de vulnerabilidad de los productores, especialmente de pequeños agricultores que dependen de este cultivo como principal fuente de ingreso.

Tabla 1.1.*Pérdidas anuales estimadas en la producción de cacao en Ecuador (2020–2023).*

Año	Producción nacional (t)	Pérdidas por moniliasis (t)	Pérdidas por fitóftora (t)	Pérdida total (t)	Impacto económico (USD millones)
2020	340,000	102,000	51,000	153,000	368.5
2021	355,000	106,500	53,250	159,750	382.7
2022	365,000	109,500	54,750	164,250	395.8
2023	370,000	111,000	55,500	166,500	401.0

Los datos presentados en la Tabla 1.1 muestran la magnitud de las pérdidas anuales en la producción de cacao en Ecuador entre 2020 y 2023. Se observa que la moniliasis representa la mayor proporción de pérdidas, mientras que la fitóftora contribuye con un porcentaje menor pero significativo. La combinación de ambas enfermedades ha generado un impacto económico acumulado que supera los 400 millones de USD en 2023, evidenciando la alta vulnerabilidad del sector productor ante estas plagas.

El impacto de estas pérdidas no se limita al plano nacional, sino que repercute en la oferta mundial de cacao fino de aroma, del cual Ecuador es el principal exportador (ICCO, 2024). La disminución en la calidad y cantidad exportable genera efectos directos en la competitividad frente a países como Costa de Marfil, Ghana y Nigeria, que aunque también enfrentan plagas similares, cuentan con mayores recursos de mitigación.

1.4.3 Modelos de series temporales

Modelos ARMA y ARIMA

El modelo ARMA (Box et al. 2016) describe una serie temporal estacionaria como una combinación lineal de sus propios rezagos y de rezagos del error. Se denota como ARMA(p,q), el modelo está dado por:

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t \quad , \quad (1.1)$$

donde

- X_t : valor de la serie en el tiempo t .
- ϕ_i : coeficientes de la parte autoregresiva (AR).
- θ_j : coeficientes de la parte de medias móviles (MA).
- ε_j : error aleatorio en t , con media cero y varianza constante.

Cuando la serie no es estacionaria, se utiliza el modelo ARIMA, el cual incluye un componente de diferenciación d para eliminar tendencias:

$$Y_t = \nabla^d X_t \Rightarrow \text{ARIMA}(p,q,d). \quad (1.2)$$

Este modelo ha sido implementado en diversos estudios sobre cultivos tropicales, incluyendo el cacao, mostrando alta capacidad para adaptarse a patrones estacionales y fluctuaciones mensuales (Martínez et al. 2019).

Modelos SARMA y SARIMA

Los modelos estacionales SARMA y SARIMA (Hyndman and Athanasopoulos 2021), extienden los anteriores al incluir componentes estacionales. Se representan como:

$ARIMA(p,d,q)(P,D,Q)_s$ donde, P,D,Q son los órdenes de los componentes estacionales y s es la periodicidad. La expresión general es:

$$\Phi_P(B^s)\phi_p(B)\nabla^d\nabla_s^D X_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \quad , \quad (1.3)$$

donde

- B : operador de rezago ($BX_t = X_{t-1}$).
- ∇^d : diferenciación no estacional.
- ∇_s^D : diferenciación estacional.

El modelo ha demostrado ser particularmente eficaz para modelar series de producción con alta estacionalidad, como es el caso del cacao en regiones tropicales, donde la producción varía cíclicamente a lo largo del año.

En Brasil, Silva et al. (2020) implementaron un modelo SARIMA para la predicción de la producción de maíz, integrando datos de precipitación y temperatura. La precisión del modelo mejoró un 15% al incluir variables climáticas respecto a una proyección puramente univariada.

Modelos ARIMAX y SARIMAX

Los modelos ARIMAX y SARIMAX permiten incorporar covariables exógenas Z_t relevantes para explicar el comportamiento de X_t . La estructura general del ARIMAX es

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \beta Z^t + \varepsilon_t \quad , \quad (1.4)$$

donde

- X_t : valor de la serie en el tiempo t .
- ϕ_i : coeficientes de la parte autoregresiva (AR).
- θ_j : coeficientes de la parte de medias móviles (MA).
- ε_j : error aleatorio en t , con media cero y varianza constante.
- βZ^t : componente de las covariables exógenas.

En el caso del SARIMAX se integran los componentes estacionales anteriormente descritos. Estos modelos son útiles cuando la producción de cacao está influenciada por factores como la precipitación, la temperatura, la incidencia de enfermedades, etc. (Wei, 2006)

Su aplicación ha sido documentada por Torres and Jiménez (2020), quienes modelaron la producción de cacao en la provincia de Esmeraldas usando datos mensuales de precipitación y temperatura como predictores en un modelo SARIMAX.

En India, Patel and Shukla (2021) emplearon modelos ARIMAX para estimar la producción de arroz, incorporando series de precipitación monzónica. Encontraron que los modelos con covariables explicativas ofrecieron intervalos de confianza más estrechos y menor error absoluto medio (MAE) que los modelos ARIMA básicos.

Modelos VARMA y VARMAX

Los modelos VARMA y VARMAX (Lütkepohl 2005), son extensiones multivariantes que permiten modelar simultáneamente varias series interrelacionadas. Un modelo VARMA(p,q) está definido por:

$$Y_t = c + \sum_{i=1}^p \Phi_i Y_{t-i} + \sum_{j=1}^q \Theta_j \varepsilon_{t-j} + \varepsilon_t, \quad (1.5)$$

donde:

- Y_t : vector de variables endógenas en t.
- Φ_i : matrices de coeficientes AR.
- Θ_j : matrices de coeficientes MA.

Cuando se añade una matriz de covariables X_t se convierte en un modelo VARMAX:

$$Y_t = c + \sum_{i=1}^p \Phi_i Y_{t-i} + \sum_{j=1}^q \Theta_j \varepsilon_{t-j} + B X_t + \varepsilon_t \quad . \quad (1.6)$$

Este modelado es útil en el estudio ya que permite explorar relaciones complejas entre producción, clima y sanidad vegetal. Estudios como el de Carrillo et al. (2021) han utilizado modelos VARMAX para evaluar la influencia cruzada entre clima y rendimiento de cultivos en la región de la costa ecuatoriana.

En Ghana, Boateng et al. (2022) aplicaron modelos VAR para estudiar la interdependencia entre la producción de cacao, la precipitación y la temperatura. El peso de la variable climática se ve evidenciado, ya que se concluyó que la precipitación acumulada de los tres meses previos explicó más del 40% de la variabilidad en la producción.

Modelo Prophet

El modelo Prophet, desarrollado por Facebook Taylor and Letham (2018) es un modelo aditivo diseñado para series temporales con fuertes componentes estacionales y tendencias no lineales. Su formulación general es:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad , \quad (1.7)$$

donde:

- $g(t)$: tendencia a largo plazo (puede ser lineal o logística).
- $s(t)$: componente estacional capturado mediante funciones trigonométricas (Fourier).
- $h(t)$: efectos de días festivos o eventos especiales (opcional).
- ε_t : término de error.

El modelo permite incorporar covariables externas (regresores adicionales) para mejorar el ajuste del modelo, lo que lo hace comparable con modelos como ARIMAX y SARIMAX. Se caracteriza por ser robusto ante datos faltantes, cambios de tendencia y valores atípicos.

Su uso permite modelar el comportamiento de la producción considerando tanto la estacionalidad como factores exógenos como precipitaciones o enfermedades, siendo una alternativa frente a métodos clásicos.

Taylor and Letham (2018) desarrollaron el modelo Prophet, aplicado a series temporales agrícolas en Estados Unidos y Europa, mostrando una capacidad robusta para capturar estacionalidad múltiple y tendencias no lineales. Su uso en cultivos sensibles al clima se ha visto incentivado debido a la mejora en la precisión de las proyecciones en comparación con ARIMA.

1.4.4 Evaluación de Modelos

La selección y comparación de modelos se basa en varias métricas que permiten evaluar la precisión, sesgo y capacidad explicativa del modelo de pronóstico. Entre las más utilizadas se encuentran:

- **ME (Mean Error):** Muestra si el modelo tiende a sobrestimar o por el contrario, subestimar.
- **RMSE (Root Mean Squared Error):** Penaliza errores grandes, mide la precisión general del modelo.
- **MAE (Mean Absolute Error):** Es menos sensible a valores atípicos en comparación con el RMSE.
- **MPE (Mean Percentage Error):** Es útil para analizar el sesgo de manera relativa.
- **MAPE (Mean Absolute Percentage Error):** Permite realizar comparaciones entre diferentes modelos con distintas escalas.
- **MASE (Mean Absolute Scaled Error):** Escala el MAE en relación a un modelo simple, facilitando así una evaluación del rendimiento relativa.
- **ACF1 (Autocorrelation of residuals at lag 1):** Determina la autocorrelación de los residuos para el primer rezago; idealmente, su valor debe estar cerca de cero para prevenir errores sistemáticos.
- **Theil's U:** Es una métrica relativa que analiza el rendimiento del modelo comparado con una predicción simple; valores por debajo de 1 indican un mejor desempeño que el modelo ingenuo.

- **AIC (Akaike Information Criterion) y BIC (Bayesian Information Criterion):** Analizan la relación entre el ajuste del modelo y su complejidad; se consideran preferibles los valores más bajos.

En este estudio, se prioriza la selección del modelo con menor RMSE y MAPE. En caso de obtener resultados similares entre modelos, se recurre a criterios de información (AIC y BIC) y métricas escaladas como MASE y Theil's U para la decisión final (Burnham and Anderson, 2004).

1.4.5 Prueba de Dickey-Fuller Aumentada (ADF)

La prueba de Dickey-Fuller Aumentada es un método estadístico esencial en el análisis de series temporales para determinar si una serie contiene una raíz unitaria, es decir, si es no estacionaria, lo cual puede llevar a estimaciones inválidas en modelos econométricos (Ramírez 2020).

Base teórica

La ADF (Dickey and Fuller 1979) extiende la prueba de Dickey-Fuller clásica incorporando términos retardados de la variable dependiente para corregir la autocorrelación en los errores. Su forma general es:

$$\Delta y_t = \alpha + \beta t + \gamma y_t + \sum_{i=1}^{p-1} \delta_i \Delta y_{t-i} + \epsilon_t,$$

donde:

- Δy_t es la primera diferencia de la serie.
- α es constante.
- βt es la tendencia.

- p es el número de rezagos.
- δ_i corrigen autocorrelación.

Bajo la hipótesis nula ($\mathbf{H}_0 : \gamma = 0$), la serie presenta raíz unitaria, es decir, no es estacionaria. Si no se rechaza que ($\mathbf{H}_0 : \gamma < 0$), se concluye que la serie es estacionaria (Ramírez 2020). La selección del orden p puede realizarse mediante criterios estadísticos como AIC o BIC, o probando la significancia de los coeficientes hacia atrás (Cheung and Lai, 1995).

Una vez estimado el modelo, el estadístico principal es el coeficiente γ . Si su estadístico t es menor (más negativo) que el valor crítico, se rechaza $\mathbf{H}_0$ y se concluye que la serie es estacionaria. En caso contrario, se suelen aplicar transformaciones como diferenciación para lograr estacionariedad (Ramírez 2020).

CAPÍTULO 2

2. METODOLOGÍA

2.1 Formulacion de alternativas

Con el objetivo de generar pronósticos precisos sobre la producción de cacao en Ecuador, se evaluaron diversas metodologías estadísticas y computacionales. Se consideraron los siguientes enfoques modelísticos, previamente descritos en el marco teórico:

- Modelo ARIMA y SARIMA.
- Modelos con variables exogenas: ARIMAX y VARMAX.
- Modelos estacionales con ruido blanco: SARMA y SARIMAX.
- Modelos aditivos con múltiples estacionalidades y covariables: Prophet.
- Modelos de suavizamiento exponencial: ETS (Error-Trend-Seasonal).
- Metodos alternativos evaluados: regresion lineal múltiple, redes neuronales recurrentes (RNN), árboles de decisión y redes LSTM.

La elección de los modelos de series temporales mencionados se basó en la estructura de los datos (mensuales, multivariantes, con estacionalidad) y en estudios previos que validan su aplicabilidad en contextos agrícolas y climáticos.

Modelos basados en aprendizaje automático, como LSTM o XGBoost, fueron descartados debido a la necesidad de grandes volúmenes de datos para un entrenamiento adecuado, su menor interpretabilidad en contextos productivos agrícolas, y los recursos computacionales que requieren para su implementación en interfaces ligeras como R Shiny.

2.2 Diseño del modelo utilizado

El enfoque adoptado consistió en el desarrollo de un sistema interactivo en R Shiny para evaluar y comparar múltiples modelos de series temporales. El sistema permitía seleccionar una variable dependiente y varias covariables, con lo cual se alimentaban distintos modelos que eran ajustados, validados y comparados automáticamente.

La lógica del modelo siguió las siguientes fases:

1. Selección y carga de datos.
2. Selección de la variable objetivo y las covariables.
3. Construcción del conjunto de entrenamiento y prueba.
4. Ajuste de los modelos ARIMA, ARIMAX, SARIMA, SARIMAX, VARMA, VARIMAX, SARMA, SARMAX, PROPHET, ETS.
5. Evaluación mediante métricas como RMSE, MAE, AIC, BIC, MAPE, MASE, ACF1 y Theil's U.
6. Visualización de resultados y selección del mejor modelo basado en criterios técnicos.

Este enfoque fue preferido por su balance entre precisión, interpretabilidad y adaptabilidad, lo cual resultaba apropiado para contextos agrícolas donde los usuarios finales podían no ser

especialistas en estadística avanzada. Además, la implementación en R Shiny ofreció ventajas sobre plataformas alternativas como Dash (Python) o Power BI, debido a su integración directa con herramientas estadísticas avanzadas, su flexibilidad y su bajo costo de implementación Hyndman and Athanasopoulos (2021).

2.3 Preparacion de datos

Los datos históricos se recogieron por mes bajo la columna de *date* y la producción en tn (toneladas); las variables exógenas fueron:

- Temperatura promedio (°C)
- Precipitacion (mm)
- Horas de luz solar
- Incidencia de Moniliasis
- Incidencia de Phytophthora

Los datos fueron proporcionados en formato tabular. El proceso de preparación incluyó:

- Conversión de fechas en formato *Date*
- Conversión de variables categóricas y numéricas en formato estándar.
- Imputación de valores faltantes mediante interpolación lineal y forward fill.
- Detección y tratamiento de valores atípicos mediante el método de tukey y control visual.
- Transformación de datos a formato de series temporales (*ts*) con frecuencia mensual.

Se establecieron ventanas móviles para dividir el conjunto de entrenamiento y prueba bajo una lógica de validación tipo "rolling forecast" lo que permitía simular un entorno real de predicción futura y mejoraba la robustez del modelo final.

2.4 Especificaciones técnicas

El sistema final fue implementado en R v4.3.0, utilizando las siguientes librerías: forecast, tseries, vars, MTS, smooth, ggplot2, shiny, shinydashboard, readxl, DT, prophet, bslib, corrplot, dplyr, lubridate, magrittr, caret, writexl, stringr, rmarkdown, pagedown, kableExtra, knitr, tinytex, tidyr.

Los modelos implementados permitieron incluir:

- Modelos ARIMA(p,d,q) y SARIMA(p,d,q)(P,D,Q)s
- Modelos ARIMAX y SARIMAX, con *xreg* como matriz de covariables.
- Modelos VARMA(p,q) y VARIMAX(p,q) multivariantes.
- Modelos SARMA y SARMAX con estacionalidad y covariables.
- Modelo ETS (Error-Trend-Seasonal), adecuado para series con componentes estructurales bien definidos.
- Modelo Prophet, desarrollado por Facebook, orientado a series con estacionalidad múltiple y cambios en tendencia.

Las estrategias de validación incluyeron:

- Separación 80/20 del conjunto de datos (entrenamiento/prueba).

- Validación cruzada mediante "rolling forecast".
- Comparación de modelos usando métricas de error y de información:
 - RMSE (Root Mean Square Error)
 - MAE (Mean Absolute Error)
 - MAPE (Mean Absolute Percentage Error)
 - MASE (Mean Absolute Scaled Error)
 - ME (Mean Error)
 - Theil's U statistic
 - ACF1 (Autocorrelación del residuo en el primer rezago)
 - AIC (Akaike Information Criterion)
 - BIC (Bayesian Information Criterion)

Además, se desarrolló una aplicación interactiva en R Shiny que permitió al usuario seleccionar covariables explicativas, cargar nuevos conjuntos de datos y visualizar gráficamente las proyecciones generadas por cada modelo. Los modelos fueron evaluados mediante las métricas previamente descritas, registrando sus resultados en una tabla comparativa integrada en la interfaz.

Para la selección del modelo óptimo se priorizó el menor valor de RMSE, complementando esta evaluación con otras métricas de precisión y consistencia. En el caso particular del modelo Prophet, se implementó un esquema de validación cruzada con ventanas móviles, acorde con su estructura basada en regresión aditiva y estacionalidades múltiples. Por otro lado, los modelos

ETS y ARIMA fueron contrastados frente a predicciones ingenuas mediante el estadístico de Theil, a fin de evaluar su capacidad predictiva relativa.

2.5 Robustez y escalabilidad

El sistema fue diseñado para adaptarse a nuevas actualizaciones de datos sin requerir reentrenamientos completos. Asimismo, podía escalarse fácilmente para incluir nuevas variables climatológicas o sanitarias del cultivo. Se implementaron controles para evitar errores de ejecución frente a datos faltantes o fuera de rango.

El código modular permite que futuros investigadores o instituciones agrícolas integren nuevos modelos o visualizaciones sin reescribir completamente el sistema.

2.6 Reproducibilidad y despliegue

El sistema puede ejecutarse localmente o desplegarse en servidores institucionales mediante *shinyapps.io* o *Rstudio Connect*, sin requerir hardware especializado. Los resultados del modelo podían exportarse, lo cual facilita la transparencia y auditoría del análisis.

2.7 Consideraciones éticas y legales

Los datos utilizados no contienen información sensible ni identificable a nivel personal. No obstante, se aplicaron las siguientes prácticas:

- Anonimización de cualquier posible identificador.
- Cumplimiento con la Ley Organica de Proteccion de Datos Personales segun Asamblea Nacional del Ecuador (2021).

- Uso exclusivo de fuentes abiertas o institucionales MAG (2022); INIAP (2020).
- Manejo controlado de archivos cargados por el usuario dentro del entorno Shiny, limitando el acceso a solo los datos utilizados para análisis estadístico y garantizando que no se almacenen fuera de la sesión activa.

Se aseguraron mecanismos de control y confidencialidad de los datos utilizados y el sistema final permite su despliegue en entornos institucionales sin comprometer información sensible. Además, el proyecto cumple con los principios de transparencia, reproducibilidad y no discriminación en el uso de modelos estadísticos.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

3.1 Descripción de los datos y preprocesamiento

El conjunto de datos estuvo conformado por la serie mensual *Producción de Cacao* (Toneladas) y un conjunto de covariables climáticas (p.ej., *Temperatura*, *Precipitación*) registradas con frecuencia mensual. La serie se modeló como una serie temporal con frecuencia 12 (`frequency = 12`) iniciando en enero de 2022, de acuerdo con la configuración usada en el aplicativo (`ts(datos$Produccion_Cacao_Toneladas, start = c(2022,1), frequency = 12)`). El tratamiento previo consideró:

- Conversión de *Fecha* a clase *Date* y renombrado a *ds* para compatibilidad con *prophet*.
- Selección de covariables estrictamente numéricas, excluyendo campos tipo fecha o índice (*Ano*, *Mes*, *Trimestre*, etc.).
- Detección y descarte de covariables constantes (varianza cero).

En la Tabla 3.1 se reportan las estadísticas descriptivas más relevantes de la variable respuesta.

Tabla 3.1.*Estadísticas descriptivas de la producción mensual de cacao (toneladas).*

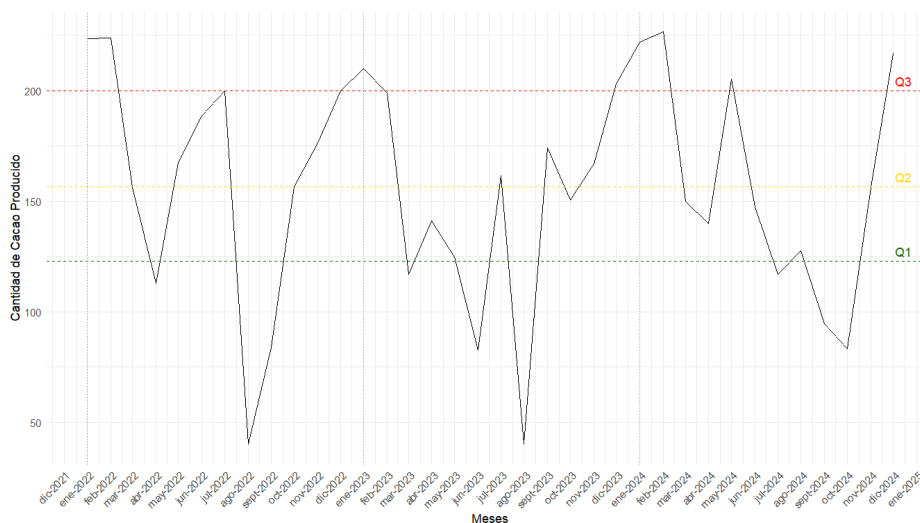
	n	Media	Mediana	Desv. estándar	Mínimo	Máximo
Producción mensual	36	155.18	156.45	51.13	40.00	226.60

3.2 Estacionariedad y estructura de dependencia

Se aplicó la prueba de Dickey–Fuller Aumentada (ADF) a la serie original para evaluar estacionariedad en nivel. El resultado del test indicó lo siguiente:

- Valor- p ADF: ≥ 0.05 .
- Criterio de decisión (umbral 0.05): la serie no era estacionaria.

Con base en lo anterior, se aplicó diferenciación estacional y/o no estacional adicional dentro de los modelos automáticos (`auto.arima` y `auto.arima` con `seasonal = TRUE`), lo que quedó reflejado en los órdenes estimados.

**Figura 3.1.***Evolutivo de producción del cacao desde el año 2022 al 2024 según meses.*

3.3 Correlación entre variables

La Tabla 3.2 muestra la matriz de correlación entre todas las variables numéricas empleadas, facilitando la detección de covariables redundantes y la selección de regresores exógenos (*xreg*) en los modelos ARIMAX - SARIMAX.

3.3.1 Variables numéricas utilizadas

- **Produccion_Cacao_Toneladas:** Esta variable muestra la producción mensual de cacao en toneladas, es la variable dependiente.
- **Temperatura_Promedio_C:** Esta covariable muestra la temperatura promedio de los meses de producción en grados celcius.
- **Precipitacion_mm:** Esta covariable muestra los niveles de precipitación mensuales en unidades de mililitros.
- **Horas_Luz_Solar:** Esta covariable indica las horas de luz solar que tuvo la producción.
- **Incidencia_Monilia:** Esta covariable indica los niveles de incidencia de monilia.
- **Incidencia_Phytophthora:** Esta covariable indica los niveles de incidencia de fitoftora.

Tabla 3.2.
Matriz de correlación entre variables numéricas.

	Produccion_Cacao Toneladas	Temperatura Promedio_C	Precipitacion_mm	Horas Luz_Solar	Incidencia Monilla	Incidencia Phytophthora	Incidencia Total	Indice Climatico
Produccion_Cacao_Toneladas	1	-0.382	-0.467	0.097	-0.620	-0.632	-0.652	-0.293
Temperatura_Promedio_C	-0.382	1	0.782	0.336	0.534	0.128	0.370	0.825
Precipitacion_mm	-0.467	0.782	1	0.677	0.242	-0.126	0.083	0.958
Horas_Luz_Solar	0.097	0.336	0.677	1	-0.293	-0.632	-0.461	0.784
Incidencia_Monilla	-0.620	0.534	0.242	-0.293	1	0.839	0.968	0.188
Incidencia_Phytophthora	-0.632	0.128	-0.126	-0.632	0.839	1	0.948	-0.246
Incidencia_Total	-0.652	0.370	0.083	-0.461	0.968	0.948	1	-0.003
Indice_Climatico	-0.293	0.825	0.958	0.784	0.188	-0.246	-0.003	1

3.4 Modelado y validación

Se estimaron y compararon los siguientes enfoques:

1. **ETS** (*Error–Trend–Seasonal*) sobre la serie `ts` de producción.
2. **ARIMA** sin regresores exógenos.
3. **SARIMA** (extensión estacional de ARIMA, con `auto.arima` y `seasonal = TRUE`).
4. **ARIMAX** con matriz `xreg` depurada (sin columnas constantes y con correlaciones definidas).
5. **SARIMAX** (`auto.arima` con `seasonal = TRUE`, prueba OCSB para estacionalidad y regresores exógenos).
6. **PROPHET** con estacionalidad anual aditiva y *changepoints* controlados, incorporando las covariables mediante `add_regressor`.
7. **VAR** (*Vector AutoRegression*), considerando la producción y sus covariables como un sistema multivariado dinámico.

Especificación y validación en PROPHET

Para PROPHET, se implementó validación deslizante con ventana de entrenamiento de 24 meses e iteraciones múltiples, promediando las métricas por iteración. La configuración principal fue:

`yearly.seasonality = TRUE, weekly = FALSE, daily = FALSE, seasonality.mode = additive,`

`n.changepoints = 10, changepoint.prior.scale = 0.05, seasonality.prior.scale = 10.`

En cada iteración se calcularon *ME*, *RMSE*, *MAE*, *MPE*, *MAPE*, *MASE*, *ACF1* y el estadístico *Theil's U*, promediando al final para obtener un desempeño estable.

3.5 Comparativo de desempeño

La Tabla 3.3 recopila las métricas de evaluación obtenidas para cada modelo, siguiendo el patrón de presentación de criterios comparativos del documento (estructura de columnas y formato booktabs). La selección del mejor modelo se realizó minimizando *RMSE*; en caso de empates prácticos, se consideró *MAPE* y *Theil's U* como criterios secundarios.

Tabla 3.3.
Comparativo de modelos (métricas promedio).

Modelo	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil_U
ETS	-27.1573	52.0769	46.2549	-33.1167	41.9174	0.9945	0.2511	1.1438
ARIMA	-27.1533	52.0748	46.2522	-33.1133	41.9146	0.9945	0.2511	1.1437
SARIMA	-61.2839	83.3857	75.6615	-59.6305	70.8982	1.6268	-0.3346	1.8314
ARIMAX	-17.1993	24.9089	18.8422	-12.2436	13.8704	0.4051	0.2374	0.5471
SARIMAX	-54.1076	62.0779	54.1076	-44.6354	44.6354	1.1633	0.2868	1.3634
PROPHET	2.3837	129.7294	102.1072	14.1032	103.4528	2.1954	-0.3050	2.8492
VAR	-39.1406	153.8404	135.6829	-32.1505	110.2649	2.9173	-0.1673	3.3788

Con base en la Tabla 3.3 el modelo con menor *RMSE* fue ARIMAX. En términos de sesgo promedio (*ME*) y eficiencia relativa (*Theil's U*), ARIMAX ventajas adicionales respecto a las alternativas.

3.6 Pronósticos

Trayectoria del mejor modelo

La Figura 3.2 presenta la trayectoria ajustada y los pronósticos a $h = 6$ meses del modelo seleccionado, con bandas de incertidumbre al 80–95% cuando el método lo permitió.

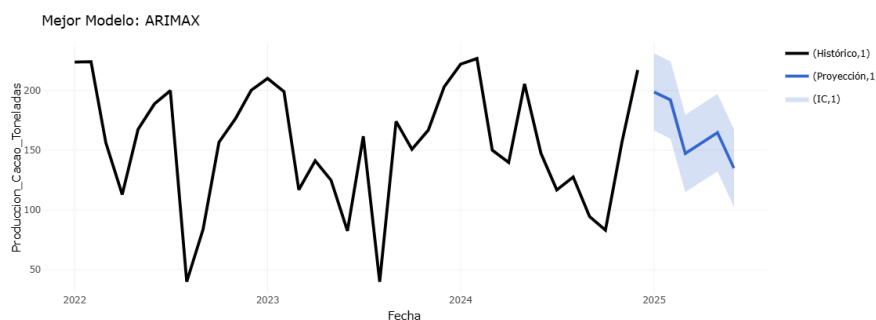


Figura 3.2.
Pronóstico del mejor modelo (ARIMAX).

Trayectoria de los modelos restantes

Las siguientes figuras: Figura 3.3, Figura 3.4, Figura 3.5, Figura 3.6, Figura 3.7, muestran las trayectorias de los modelos que no resultaron óptimos según la evaluación de las métricas.

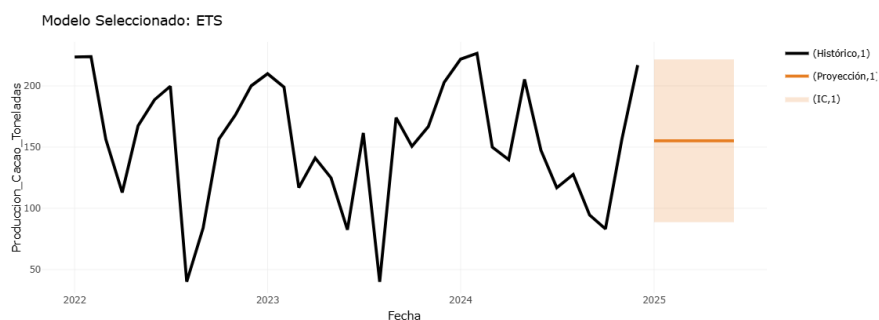


Figura 3.3.
Pronóstico del modelo ETS.

En términos generales, el modelo ETS presentó pronósticos muy estables y prácticamente constantes en torno a los 155, mostrando alta estabilidad pero poca capacidad para adaptarse a variaciones estacionales o cambios abruptos de la serie. Esto se refleja en métricas de error moderadas ($RMSE = 52.08$, $MAE = 46.25$), lo que indica que, si bien sus predicciones son consistentes, tienden a rezagarse frente a las fluctuaciones reales observadas en la serie.

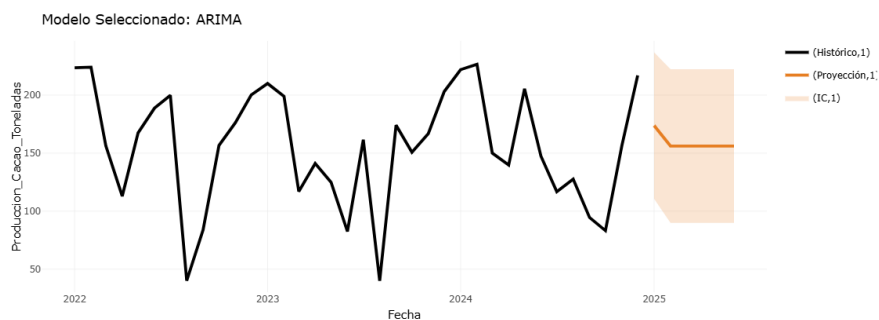


Figura 3.4.
Pronóstico del modelo ARIMA.

El modelo ARIMA mostró un comportamiento muy similar al de ETS, con pronósticos estables y prácticamente constantes en torno a 156. Aunque mantiene coherencia con la tendencia general, no logra capturar adecuadamente las fluctuaciones ni los picos de la serie, especialmente en meses como junio, lo que se refleja en errores promedio casi idénticos a los de ETS ($RMSE = 52.07$, $MAE = 46.25$).

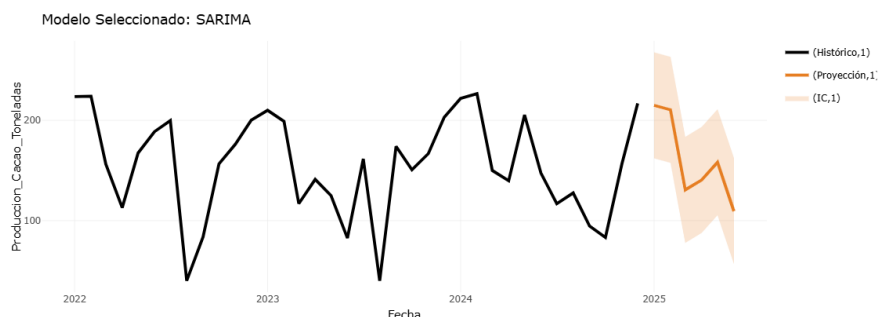


Figura 3.5.
Pronóstico del modelo SARIMA.

SARIMA mostró un desempeño inferior a ARIMA, con pronósticos menos precisos y errores promedio significativamente mayores ($RMSE = 83.39$, $MAE = 75.66$). Aunque intenta capturar la estacionalidad, el modelo no logra reflejar adecuadamente los picos de ciertos meses, mostrando limitaciones para predecir la variabilidad mensual.

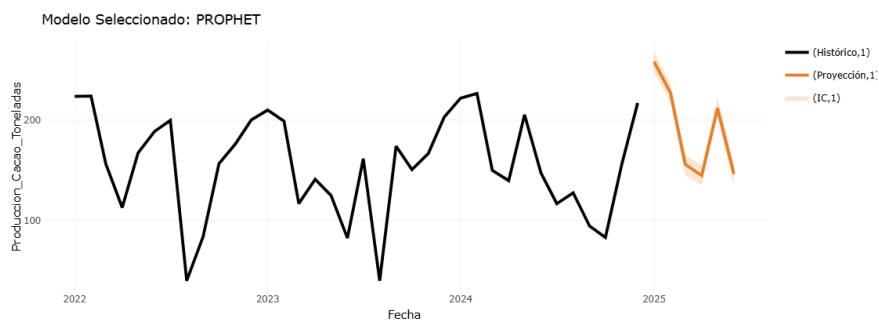


Figura 3.6.
Pronóstico del modelo PROPHET.

El modelo Prophet evidenció sobreajuste, con pronósticos más altos que los de los demás

modelos en enero, febrero y junio. Aunque presenta baja autocorrelación residual ($ACF1 = -0.305$), su tendencia a exagerar los picos indica que no captura correctamente la estacionalidad ni los efectos de covariables, generando errores elevados ($RMSE = 129.73$, $MAE = 102.11$).

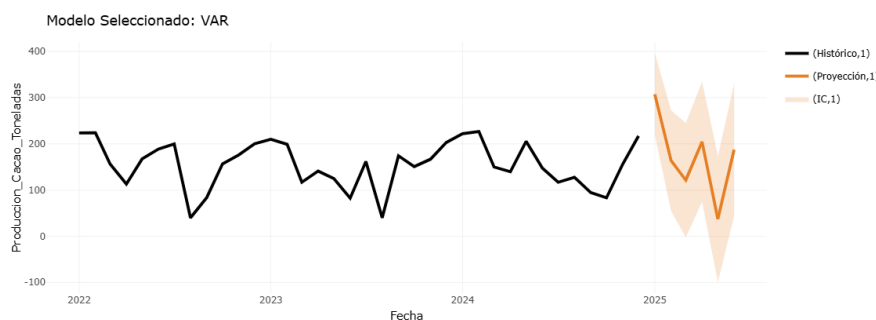


Figura 3.7.
Pronóstico del modelo VAR.

El modelo VAR mostró un desempeño limitado: sigue la tendencia general de la serie, pero sus pronósticos se alejan considerablemente del comportamiento real en meses extremos, especialmente en junio y diciembre, reflejando errores elevados ($RMSE = 153.84$, $MAE = 135.68$).

Comparación visual de pronósticos

Para complementar la comparación cuantitativa, la Figura 3.8 muestra en un mismo panel los pronósticos de todos los modelos frente a la serie histórica, facilitando la inspección de diferencias en nivel, tendencia y amplitud estacional entre métodos.

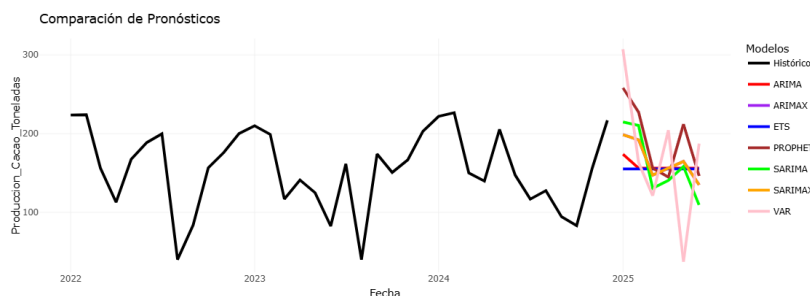


Figura 3.8.
Comparación de pronósticos: ETS, ARIMA, ARIMAX, SARIMAX, PROPHET, SARIMA, VAR.

3.7 Estimación de ganancias económicas mensuales

El modelo ARIMAX permitió obtener una proyección de la producción mensual de cacao para el año 2025. A partir de estas proyecciones se calcularon las ganancias económicas potenciales; para ello se tomó como referencia el precio promedio histórico de exportación, aproximadamente 2,406 dólares por tonelada.

Tabla 3.4.

Producción proyectada y valor económico mensual (ARIMAX, 2025)

Mes	Producción (t)	Valor económico (USD)
Enero 2025	198.6	477,866.4
Febrero 2025	192.0	461,860.2
Marzo 2025	147.2	354,202.4
Abril 2025	156.2	375,781.6
Mayo 2025	164.7	396,223.8
Junio 2025	135.0	324,852.0

De acuerdo con la Tabla 3.4, se evidencia que, en los primeros seis meses del 2025, la producción proyectada varía entre 135 y 199 toneladas. En términos económicos, esto representa un rango mensual de aproximadamente 325 mil a 478 mil dólares.

3.8 Discusión de resultados

El proyecto mostró que los modelos ARIMAX y SARIMAX proporcionaron los pronósticos más precisos, superando a los enfoques univariantes tradicionales como lo son ARIMA o

SARIMA y a métodos base como ETS. Esto refleja el valor de incorporar covariables climáticas en la predicción como muestran Muñoz et al. (2022), la inclusión de factores exógenos mejoró notablemente la exactitud del modelo.

Los hallazgos coinciden con los de Silva et al. (2020), quienes obtuvieron un incremento del 15% en la precisión al añadir datos de lluvia y temperatura en la predicción de cultivos.

De igual modo, se evidenció que la precipitación fue una variable determinante en la dinámica de las enfermedades del cacao, en línea con Boateng et al. (2022) que atribuyen a la lluvia acumulada un peso explicativo superior al 40% de la variabilidad en la producción de cacao. Esta convergencia muestra que los modelos multivariantes mantienen relaciones importantes entre clima y agricultura, tal como señalan Pinto et al. (2015) al encontrar correlaciones significativas entre el rendimiento del cacao y las condiciones climáticas en Brasil. La aproximación multivariable de nuestro proyecto integrando clima y sanidad del cultivo se asemeja al enfoque de Galarza et al. (2021) y Carrillo et al. (2021), quienes aplicaron modelos VAR/VARMAX para representar interdependencias entre clima, sanidad vegetal y producción de cacao. Sin embargo, este proyecto, a diferencia de los trabajos mencionados, no se enfoca únicamente en la producción o rendimiento, sino que además aborda directamente las enfermedades fúngicas como variable de interés, aportando una perspectiva novedosa centrada en la alerta fitosanitaria temprana.

Por otro lado, el algoritmo Prophet mostró tendencia al sobreajuste en nuestros datos; esto se explica por un alto error a pesar de la baja autocorrelación residual, mientras que en otros contextos agrícolas Prophet ha demostrado capturar eficazmente patrones estacionales complejos con alta robustez.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

A partir de los objetivos planteados, la metodología aplicada y los resultados obtenidos, se presentan las siguientes conclusiones:

- El desarrollo del presente proyecto permitió automatizar la construcción, validación y comparación de modelos de series temporales multivariantes, logrando consolidar una herramienta innovadora en R Shiny que integra variables climáticas y fitosanitarias para el pronóstico de enfermedades fúngicas en el cultivo de cacao. Este avance representa un aporte significativo, tanto a nivel académico como productivo, al poner a disposición de investigadores y productores una plataforma accesible, escalable y reproducible.
- Los resultados evidenciaron que los modelos con variables exógenas, particularmente ARIMAX, superan de manera notable a los modelos univariantes en términos de precisión, confirmando que la inclusión de factores como la precipitación y la incidencia de enfermedades incrementa de forma sustancial la calidad de los pronósticos.
- Se demostró que la estacionalidad y la no estacionariedad presentes en la producción de

cacao afectan la capacidad predictiva de los modelos más simples (ETS, ARIMA), mientras que enfoques multivariantes permiten capturar mejor la dinámica del sistema. El modelo Prophet, aunque flexible, presentó sobreajuste y menor capacidad de generalización en este contexto.

- Desde el punto de vista económico, el uso de modelos predictivos multivariantes representa una contribución valiosa para reducir pérdidas derivadas de la moniliasis y la fitóftora, que en Ecuador alcanzan entre el 30% y 40% de la producción anual. Anticipar brotes permite optimizar la asignación de recursos, mejorar la toma de decisiones en campo y contribuir a la sostenibilidad del sector cacaotero.
- En conjunto, el proyecto constituye una contribución académica y aplicada de alto impacto, al demostrar que la matemática, la estadística y la ciencia de datos pueden integrarse exitosamente para dar respuesta a problemas reales de la agricultura ecuatoriana.

Recomendaciones

En función de los hallazgos obtenidos y con el objetivo de fortalecer el impacto del trabajo, se tienen las siguientes recomendaciones:

- Ampliar la cobertura temporal y geográfica de los registros, integrando series más extensas y datos provenientes de distintas zonas productoras del país. Esto permitirá tener una mayor robustez en los modelos, evaluar cambios estructurales y mejorar la confiabilidad de los pronósticos.
- Incorporar módulos que integren datos oficiales de predicción meteorológica con las proyecciones del modelo, generando alertas de riesgo en tiempo real que faciliten la planificación de medidas preventivas y la gestión eficiente de recursos en el campo.
- Adaptar el enfoque multivariante a la gestión de enfermedades en productos agrícolas de alta relevancia como el banano, el café y el arroz, con el fin de ampliar el impacto de la investigación y contribuir a la seguridad alimentaria y al desarrollo agroexportador del Ecuador.
- Promover la interacción entre matemáticos, agrónomos, fitopatólogos y expertos en ciencia de datos, fortaleciendo así la innovación y asegurando que la herramienta evolucione en línea con las necesidades reales del sector agrícola.

BIBLIOGRAFÍA

Asamblea Nacional del Ecuador (2021). Ley orgánica de protección de datos personales.

Boateng, K., Mensah, D., and Owusu, J. (2022). Modeling cocoa production in ghana using vector autoregressive (var) models with climatic covariates. *African Journal of Agricultural Economics*, 15(1):57–73.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition.

Burnham, K. P. and Anderson, D. R. (2004). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, 2 edition.

Carrillo, F., Maldonado, D., and Mero, L. (2021). Análisis multivariado del clima y rendimiento del cacao en la costa ecuatoriana. *Revista Científica Agrovida*, 8(1):54–63.

Cheung, Y. and Lai, K. (1995). Lag order and the finite-sample behavior of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431.

FAO (2025). Faostat: Crops and livestock products (qcl) — cocoa beans: Production (ecuador). Dataset; accessed 18 August 2025.

- Galarza, V., Moreira, A., and Pazmiño, R. (2021). Modelos var para el pronóstico de exportaciones de cacao. *Revista Equinoccio*, 18(2):15–27.
- García, L. (2015). Impacto de la moniliasis en la producción de cacao en México. Scribd. Accedido el 18 de agosto de 2025.
- Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts, 3 edition. <https://otexts.com/fpp3/>.
- ICCO (2024). Quarterly bulletin of cocoa statistics: Issue no. 4 — volume I (cocoa year 2023/24).
- INIAP (2020). Moniliasis del cacao.
- Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. Springer.
- MAG (2021). Panorama agroeconómico del Ecuador 2021. Accedido el 18 de agosto de 2025.
- MAG (2022). Informe de enfermedades fúngicas en cultivos de cacao en Ecuador.
- Martínez, D., Castillo, M., and Luján, C. (2019). Análisis de la producción de cacao mediante series temporales. *Revista de Ciencias Agropecuarias*, 36(2):112–122.
- Martínez, S., López, R., and Hernández, A. (2017). Incidencia de la moniliasis en plantaciones de cacao en Nicaragua. *Revista de Ciencias Agrícolas*, 9(2):77–89. Accedido el 18 de agosto de 2025.
- Muñoz, P., Ruales, S., and Paredes, M. (2022). Uso de modelos SARIMA y ARIMAX para el pronóstico de la producción de cacao. *Revista Ecuatoriana de Estadística*, 11(1):34–48.
- Patel, A. and Shukla, N. (2021). ARIMAX modeling for rice yield prediction in India using monsoon rainfall. *Journal of Agricultural Statistics*, 76(3):221–238.

- Pinto, A., Carvalho, M., and Santos, R. (2015). Forecasting cocoa production in brazil using arima models. *Journal of Agricultural Economics Research*, 58(4):245–257.
- Ramírez, W. B. (2020). Fundamentos para pronosticar una serie de tiempo estacionaria con información de su propio pasado. *Sistemas e Informática Industrial*, 23(1):209–214. Accedido el 18 de agosto de 2025.
- Rodríguez, J. and Pérez, C. (2019). Efectos de la moniliasis en la producción de cacao en colombia. *Revista de Producción Agrícola*, 12(3):45–59. Accedido el 18 de agosto de 2025.
- Silva, R., Santos, M., and Oliveira, P. (2020). Aplicación de modelos sarima en la predicción de la producción de maíz en brasil. *Revista Brasileira de Estatística Aplicada*, 34(2):115–132.
- Taylor, S. J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45.
- Torres, J. and Jiménez, L. (2020). Pronóstico de la producción de cacao en esmeraldas usando modelos sarimax. *Revista Científica Tierra Fértil*, 10(1):77–85.
- Wei, W. W. S. (2006). *Time Series Analysis: Univariate and Multivariate Methods*. Pearson, 2 edition.

APÉNDICES

APÉNDICE A

```
1 # =====
2 # App Shiny: Pronóstico de Producción de Cacao
3 # =====
4
5 # ---- 1) Paquetes ----
6 req_pkgs <- c(
7   "shiny", "bslib", "DT", "plotly", "ggplot2", "dplyr", "lubridate", "magrittr",
8   "forecast", "tseries", "caret", "writexl", "prophet", "vars", "stringr", "
9   "rmarkdown", "pagedown", "kableExtra",
10  "knitr", "tinytex", "tidyr"
11 )
12 new_pkgs <- req_pkgs[!(req_pkgs %in% installed.packages()[, "Package"])]
13 if (length(new_pkgs)) install.packages(new_pkgs, dependencies = TRUE)
14 invisible(lapply(req_pkgs, library, character.only = TRUE))
15
16 # ---- 2) Helpers ----
17
18 # Detecta separador leyendo primeras líneas (coma vs punto y coma)
19 detectar_sep <- function(path) {
20   head_txt <- tryCatch(readLines(path, n = 5, warn = FALSE), error =
21     function(e) "")
```

```

20   if (length(head_txt) == 0) return(",")
21   comas <- sum(stringr::str_count(head_txt, ","))
22   pcomas <- sum(stringr::str_count(head_txt, ";"))
23   if (pcomas > comas) ";" else ","
24 }
25
26 # Convierte columnas a numérico (acepta "1.234,56" y "1,234.56")
27 to_numeric_safe <- function(x) {
28   if (is.numeric(x)) return(x)
29   if (inherits(x, "Date")) return(as.numeric(x))
30   x <- as.character(x)
31   x <- gsub("\\s+", "", x)
32   # tomar un ejemplo no vacío para heurística
33   idx <- which(!is.na(x) & nzchar(x))[1]
34   sample_val <- if (!is.na(idx)) x[idx] else ""
35   # Si hay coma y punto, asumimos formato europeo "1.234,56" -> quitamos
36   # '.' y cambiamos ',' por '.'
37   if (nzchar(sample_val) && grepl(",", sample_val) && grepl("\\.", sample_val)) {
38     x <- gsub("\\.", "", x)
39   }
40   x <- gsub(",", ".", x)
41   suppressWarnings(as.numeric(x))
42 }

```

```

43 # Construye ts mensual (freq=12) a partir de vector y fechas
44 make_ts12 <- function(y, fechas) {
45     fechas <- as.Date(fechas)
46     ord <- order(fechas)
47     fechas <- fechas[ord]; y <- y[ord]
48     start_y <- lubridate::year(min(fechas, na.rm = TRUE))
49     start_m <- lubridate::month(min(fechas, na.rm = TRUE))
50     ts(y, frequency = 12, start = c(start_y, start_m))
51 }
52
53 # Métricas uniformes + Theil's U, ACF1, MASE
54 compute_metrics <- function(actual, pred, insample=NULL) {
55     actual <- as.numeric(actual); pred <- as.numeric(pred)
56     err <- actual - pred
57     me <- mean(err, na.rm = TRUE)
58     rmse <- sqrt(mean(err^2, na.rm = TRUE))
59     mae <- mean(abs(err), na.rm = TRUE)
60     mpe <- mean(err/actual, na.rm = TRUE) * 100
61     mape <- mean(abs(err/actual), na.rm = TRUE) * 100
62     if (!is.null(insample) && length(insample) > 1) {
63         mae_naive <- mean(abs(diff(insample)), na.rm = TRUE)
64     } else {
65         mae_naive <- mean(abs(diff(actual)), na.rm = TRUE)
66     }

```

```

67 mase <- ifelse(is.finite(mae_naive) && mae_naive > 0, mae/mae_naive, NA_
    real_)
68 acf1 <- tryCatch(acf(err, plot = FALSE)$acf[2], error = function(e) NA_
    real_)
69 if (length(actual) >= 2) {
70     rmse_naive <- sqrt(mean((actual[-1] - actual[-length(actual)])^2, na.rm
        = TRUE))
71     theil_u <- ifelse(rmse_naive > 0, rmse/rmse_naive, NA_real_)
72 } else {
73     theil_u <- NA_real_
74 }
75 data.frame(ME=me, RMSE=rmse, MAE=mae, MPE=mpe, MAPE=mape, MASE=mase, ACF1
    =acf1, Theil_U=theil_u)
76 }
77
78 # Limpia xreg: quita constantes y multicolinealidad
79 clean_xreg <- function(df_num, cutoff=0.95) {
80     if (is.null(df_num) || ncol(df_num) == 0) return(df_num)
81     keep <- sapply(df_num, function(z) is.numeric(z) && sd(z, na.rm = TRUE) >
        0)
82     df_num <- df_num[, keep, drop=FALSE]
83     if (ncol(df_num) <= 1) return(df_num)
84     cor_mat <- suppressWarnings(cor(df_num, use = "pairwise.complete.obs"))
85     if (any(is.na(cor_mat))) {
86         good <- colSums(is.na(cor_mat)) == 0

```

```

87     df_num <- df_num[, good, drop=FALSE]
88     if (ncol(df_num) <= 1) return(df_num)
89     cor_mat <- cor(df_num, use = "pairwise.complete.obs")
90 }
91 rem <- tryCatch(caret::findCorrelation(cor_mat, cutoff = cutoff), error =
      function(e) integer(0))
92 if (length(rem) > 0) df_num <- df_num[, -rem, drop=FALSE]
93 df_num
94 }
95
96 # Proyección de covariables (ARIMA univar) para h meses
97 forecast_covariables <- function(df_num, h, fechas) {
98     if (is.null(df_num) || ncol(df_num) == 0) return(NULL)
99     fut_list <- list()
100     for (nm in colnames(df_num)) {
101         tsx <- make_ts12(df_num[[nm]], fechas)
102         mod <- tryCatch(forecast::auto.arima(tsx), error=function(e) NULL)
103         fut_list[[nm]] <- if (is.null(mod)) rep(NA_real_, h) else as.numeric(
            forecast::forecast(mod, h=h)$mean)
104     }
105     futuro <- do.call(cbind, fut_list)
106     colnames(futuro) <- colnames(df_num)
107     futuro
108 }
109

```

```

110 # Estandariza datos de forecast a data.frame para graficar con IC
111 df_forecast_to_plot <- function(fore_obj, fechas_hist) {
112   n_hist <- length(fechas_hist)
113   fechas_fut <- seq(from = max(fechas_hist) %m+% months(1), by = "month",
114                     length.out = length(fore_obj$mean))
115   data.frame(
116     Fecha      = c(fechas_hist, fechas_fut),
117     Valor      = c(rep(NA, n_hist), as.numeric(fore_obj$mean)),
118     Inferior   = c(rep(NA, n_hist), as.numeric(fore_obj$lower[,1])),
119     Superior   = c(rep(NA, n_hist), as.numeric(fore_obj$upper[,1])),
120     Tipo       = c(rep("Histórico", n_hist), rep("Proyección", length(fore_
121               obj$mean))))
122   )
123 }
124
125 # ---- 3) UI ----
126
127 tema_profesional <- bslib::bs_theme(
128   version      = 5,
129   bootswatch= "lumen",
130   primary      = "#2C3E50",
131   secondary    = "#18BC9C",
132   base_font    = bslib::font_google("Open Sans")
133 )
134
135 ui <- fluidPage(

```

```

133 theme = tema_profesional,
134 div(titlePanel(h2(" Pronóstico de Producción de Cacao", align = "left"))
135     ),
136 sidebarLayout(
137     sidebarPanel(
138         tags$h4(" Configuración del Análisis"),
139         fileInput("archivo", " Subir archivo CSV", accept = c(".csv")),
140         selectInput("sep", "Separador de columnas:",
141             choices = c("Auto" = "auto", "Coma (,)" = ",", "Punto y
142                 coma (;)" = ";"), selected = "auto"),
143         uiOutput("var_selector"),
144         uiOutput("cov_selector"),
145         sliderInput("h", " Meses a proyectar:", min = 3, max = 12, value =
146             6, step = 1),
147         actionButton("run", " Ejecutar Modelos", class = "btn btn-success"),
148         width = 3
149     ),
150 mainPanel(
151     tags$h4(" Test de Estacionariedad (ADF)"),
152     verbatimTextOutput("adf_out"),
153     tags$h4(" Matriz de Correlación Numérica"),
154     DT::DTOutput("cor_table"),
155     br(),
156     tabsetPanel(

```

```

154     tabPanel(" Comparativa de Modelos", DT::DTOutput("tabla_modelos"))
155     ,
156     tabPanel(" Mejor Modelo", plotlyOutput("plot_mejor")),
157     tabPanel(" Todos los Modelos", plotlyOutput("plot_todos")),
158     tabPanel(" Proyección Covariables", plotlyOutput("plot_covs")),
159     tabPanel(" Proyección Conjunta", plotlyOutput("plot_conjunto")),
160     tabPanel(" Descargas",
161         br(),
162         downloadButton("descarga_xlsx", "Descargar proyección (
163             Excel)", class = "btn btn-primary")
164         # ,
165         # br(), br(),
166         # downloadButton("descarga_pdf", "Descargar informe (PDF)
167             ", class = "btn btn-outline-primary")
168     )
169 ),
170 width = 9
171 )
172 )
173 # ---- 4) Server ----
174 server <- function(input, output, session) {
175

```

```

176 # -- 4.1 Carga de datos --
177 datos_raw <- reactive({
178   req(input$archivo)
179   sep_use <- if (input$sep == "auto") detectar_sep(input$archivo$datapath
180     ) else input$sep
181   df <- tryCatch(
182     read.csv(input$archivo$datapath, sep = sep_use, stringsAsFactors =
183       FALSE, check.names = FALSE),
184     error = function(e) {
185       alt <- if (sep_use == ",") ";" else ","
186       read.csv(input$archivo$datapath, sep = alt, stringsAsFactors =
187         FALSE, check.names = FALSE)
188     }
189   )
190   # Normaliza nombre de fecha si no viene exactamente "Fecha"
191   if (!("Fecha" %in% names(df))) {
192     fecha_col <- names(df)[grepl("fecha|date|Fecha|FECHA", names(df))]
193     if (length(fecha_col) >= 1) names(df)[match(fecha_col[1], names(df))]
194       <- "Fecha"
195   }
196   validate(need("Fecha" %in% names(df), "El CSV debe contener una columna
197     'Fecha'."))
198   df$Fecha <- as.Date(df$Fecha)
199   validate(need(!any(is.na(df$Fecha)), "No se pudieron convertir las
200     fechas. Formato recomendado: YYYY-MM-DD."))

```

```

195     df <- df[order(df$Fecha), ]
196     df
197 })
198
199 # -- 4.2 Selectores dinámicos --
200 output$var_selector <- renderUI({
201     df <- datos_raw()
202     sugerida <- if ("Produccion_Cacao_Toneladas" %in% names(df)) "
203         Produccion_Cacao_Toneladas" else NULL
204     selectInput("var_dep", "Variable de producción:", choices = setdiff(
205         names(df), "Fecha"), selected = sugerida)
206 })
207
208 output$cov_selector <- renderUI({
209     df <- datos_raw()
210     req(input$var_dep)
211     # candidatas numéricas (o convertibles)
212     num_cols <- names(df)[sapply(df, is.numeric) | sapply(df, function(z)
213         all(grepl("^[-0-9,\\.]+$", as.character(z)))) ]
214     num_cols <- setdiff(num_cols, c("Fecha", input$var_dep, "Ano", "Mes", "
215         Mes_Nombre", "Trimestre"))
216     checkboxGroupInput("covs", "Covariables:", choices = num_cols,
217         selected = intersect(num_cols, c("Temperatura_
218             Promedio_C", "Precipitacion_mm", "Incidencia_
219             Phytophthora")))

```

```

214 })
215
216 # -- 4.3 Correlación (DT) --
217 output$cor_table <- DT::renderDT({
218   df <- datos_raw()
219   req(input$var_dep)
220   cols <- unique(c(input$var_dep, input$covs))
221   if (length(cols) == 0) {
222     return(DT::datatable(data.frame(Nota="Seleccione variables"), options
223       =list(dom='t'))))
224   }
225   dn <- df[, cols, drop=FALSE]
226   dn[] <- lapply(dn, to_numeric_safe)
227   cor_mat <- suppressWarnings(cor(dn, use = "complete.obs"))
228   cor_df <- round(as.data.frame(cor_mat), 3)
229   DT::datatable(cor_df, rownames = TRUE, class='cell-border stripe hover
230     compact',
231     options = list(pageLength = 8, dom='t', scrollX=TRUE,
232       columnDefs = list(list(className='dt-
233         center', targets="_all")))) %>%
234   DT::formatStyle(
235     columns = names(cor_df),
236     `font-size` = '11px',
237     padding = '2px 4px',
238     `white-space` = 'normal'

```

```

236     )
237 })
238
239 # -- 4.4 ADF (estacionariedad) --
240 output$adf_out <- renderText({
241     df <- datos_raw(); req(input$var_dep)
242     y <- to_numeric_safe(df[[input$var_dep]])
243     ts_y <- make_ts12(y, df$Fecha)
244     res <- tryCatch(tseries::adf.test(ts_y, alternative = "stationary"),
245                     error = function(e) NULL)
246     if (is.null(res)) {
247         "El test ADF no se pudo calcular (serie demasiado corta o con NA)."
248     } else if (res$p.value < 0.05) {
249         sprintf("ADF: p-valor = %.4f → Serie estacionaria (rechaza raíz
250                 unitaria).", res$p.value)
251     } else {
252         sprintf("ADF: p-valor = %.4f → Serie NO estacionaria.", res$p.value)
253     }
254 })
255
256 # -- 4.5 Ejecución de modelos --
257 resultados <- eventReactive(input$run, {
258     df <- datos_raw(); req(input$var_dep); h <- input$h
259
260     # Dep y covariables

```

```

260 y <- to_numeric_safe(df[[input$var_dep]])
261 cov_df <- NULL
262 if (!is.null(input$covs) && length(input$covs) > 0) {
263     cov_df <- df[, input$covs, drop=FALSE]
264     cov_df[] <- lapply(cov_df, to_numeric_safe)
265     cov_df <- clean_xreg(cov_df, cutoff = 0.95)
266     if (ncol(cov_df) == 0) cov_df <- NULL
267 }
268
269 # ts y split holdout (últimos h meses)
270 ts_full <- make_ts12(y, df$Fecha)
271 n <- length(ts_full)
272 validate(need(n > h + 6, "La serie es muy corta para un holdout de h
    meses. Usa menos h o agrega más datos."))
273 ts_train <- window(ts_full, end = time(ts_full)[n - h])
274 ts_test <- window(ts_full, start = time(ts_full)[n - h + 1])
275
276 # xreg alineada
277 x_train <- x_test <- NULL
278 if (!is.null(cov_df)) {
279     x_all <- as.matrix(cov_df)
280     x_train <- x_all[1:(n - h), , drop=FALSE]
281     x_test <- x_all[(n - h + 1):n, , drop=FALSE]
282 }
283

```

```

284 # Futuro de covariables para proyección
285 futuro_cov <- forecast_covariables(cov_df, h, df$Fecha)
286
287 modelos <- list()
288 metricas <- list()
289 forecasts_full <- list()
290
291 # ---- ETS ----
292 modelos$ETS <- forecast::ets(ts_train)
293 fc_ets <- forecast::forecast(modelos$ETS, h = h)
294 metricas$ETS <- compute_metrics(ts_test, fc_ets$mean, insample = ts_
      train)
295 forecasts_full$ETS <- forecast::forecast(forecast::ets(ts_full), h = h)
296
297 # ---- ARIMA (no estacional) ----
298 modelos$ARIMA <- forecast::auto.arima(ts_train, seasonal = FALSE)
299 fc_arima <- forecast::forecast(modelos$ARIMA, h = h)
300 metricas$ARIMA <- compute_metrics(ts_test, fc_arima$mean, insample = ts
      _train)
301 forecasts_full$ARIMA <- forecast::forecast(forecast::auto.arima(ts_full
      , seasonal = FALSE), h = h)
302
303 # ---- SARIMA ----
304 modelos$SARIMA <- forecast::auto.arima(ts_train, seasonal = TRUE)
305 fc_sarima <- forecast::forecast(modelos$SARIMA, h = h)

```

```

306   metricas$SARIMA <- compute_metrics(ts_test, fc_sarima$mean, insample =
      ts_train)
307   forecasts_full$SARIMA <- forecast::forecast(forecast::auto.arima(ts_
      full, seasonal = TRUE), h = h)
308
309   # ---- ARIMAX ----
310   if (!is.null(x_train)) {
311     modelos$ARIMAX <- tryCatch(forecast::auto.arima(ts_train, xreg = x_
      train, seasonal = FALSE),
312                               error=function(e) NULL)
313     if (!is.null(modelos$ARIMAX)) {
314       fc_arimax <- tryCatch(forecast::forecast(modelos$ARIMAX, xreg = x_
      test, h = h), error=function(e) NULL)
315       if (!is.null(fc_arimax)) metricas$ARIMAX <- compute_metrics(ts_test
      , fc_arimax$mean, insample = ts_train)
316       forecasts_full$ARIMAX <- tryCatch({
317         mod_full <- forecast::auto.arima(ts_full, xreg = as.matrix(cov_df
      ))
318         forecast::forecast(mod_full, xreg = futuro_cov, h = h)
319       }, error=function(e) NULL)
320     }
321   }
322
323   # ---- SARIMAX ----
324   if (!is.null(x_train)) {

```

```

325 modelos$SARIMAX <- tryCatch(
326   forecast::auto.arima(ts_train, xreg = x_train, seasonal = TRUE,
327     stepwise = FALSE, approximation = FALSE),
328   error=function(e) NULL
329 )
330 if (!is.null(modelos$SARIMAX)) {
331   fc_sarimax <- tryCatch(forecast::forecast(modelos$SARIMAX, xreg = x
332     _test, h = h), error=function(e) NULL)
333   if (!is.null(fc_sarimax)) metricas$SARIMAX <- compute_metrics(ts_
334     test, fc_sarimax$mean, insample = ts_train)
335   forecasts_full$SARIMAX <- tryCatch({
336     mod_full <- forecast::auto.arima(ts_full, xreg = as.matrix(cov_df
337       ), seasonal = TRUE, stepwise = FALSE, approximation = FALSE)
338     forecast::forecast(mod_full, xreg = futuro_cov, h = h)
339   }, error=function(e) NULL)
340 }
341 }
342
343 # ---- Prophet (+ regressors si hay) ----
344 df_prophet <- data.frame(ds = df$Fecha, y = as.numeric(y))
345 if (!is.null(cov_df)) df_prophet <- cbind(df_prophet, cov_df)
346
347 fechas <- df$Fecha
348 corte <- max(fechas) %m-% months(h)
349 train_idx <- which(fechas <= corte)

```

```

346 test_idx <- which(fechas > corte)
347 df_train <- df_prophet[train_idx, , drop=FALSE]
348 df_test <- df_prophet[test_idx, , drop=FALSE]
349
350 m <- prophet::prophet(
351   yearly.seasonality = TRUE,
352   weekly.seasonality = FALSE,
353   daily.seasonality = FALSE,
354   seasonality.mode = "additive",
355   changepoint.prior.scale = 0.05,
356   seasonality.prior.scale = 10,
357   n.changepoints = 10
358 )
359 if (!is.null(cov_df)) for (nm in colnames(cov_df)) m <- prophet::add_
    regressor(m, nm)
360 m <- prophet::fit.prophet(m, df_train)
361
362 future_test <- df_test[, c("ds", setdiff(names(df_test), c("y"))), drop
    =FALSE]
363 pred_test <- predict(m, future_test)
364 #pred_test <- prophet::predict(m, future_test)
365 metricas$PROPHET <- compute_metrics(df_test$y, pred_test$yhat, insample
    = df_train$y)
366
367 # proyección full con Prophet

```

```

368 m_full <- prophet::prophet(
369   yearly.seasonality = TRUE, weekly.seasonality = FALSE, daily.
        seasonality = FALSE,
370   seasonality.mode = "additive", changepoint.prior.scale = 0.05,
371   seasonality.prior.scale = 10, n.changepoints = 10
372 )
373 if (!is.null(cov_df)) for (nm in colnames(cov_df)) m_full <- prophet::
        add_regressor(m_full, nm)
374 m_full <- prophet::fit.prophet(m_full, df_prophet)
375 future_full <- data.frame(ds = seq(from = max(df$Fecha) %m+% months(1),
        by = "month", length.out = h))
376 if (!is.null(futuro_cov)) future_full <- cbind(future_full, as.data.
        frame(futuro_cov))
377 pred_full <- predict(m_full, future_full)
378 #pred_full <- prophet::predict(m_full, future_full)
379 forecasts_full$PROPHET <- list(
380   mean = ts(pred_full$yhat, frequency = 12),
381   lower = cbind(`80%` = pred_full$yhat_lower, `95%` = pred_full$yhat_
        lower),
382   upper = cbind(`80%` = pred_full$yhat_upper, `95%` = pred_full$yhat_
        upper)
383 )
384
385 # ---- VAR (si hay 1 covariable) ----
386 if (!is.null(cov_df) && ncol(cov_df) >= 1) {

```

```

387 mat <- cbind(y = as.numeric(y), as.data.frame(cov_df))
388
389 mat[] <- lapply(mat, to_numeric_safe)
390
391 sel <- tryCatch(vars::VARselect(mat[1:(n-h)], , lag.max = 6, type = "
    const"), error=function(e) NULL)
392
393 p_sel <- if (!is.null(sel) && "AIC(n)" %in% names(sel$selection)) sel
    $selection[["AIC(n)"]] else 2
394
395 var_mod <- tryCatch(vars::VAR(mat[1:(n-h)], , p = p_sel, type = "
    const"), error=function(e) NULL)
396
397 if (!is.null(var_mod)) {
398   var_fc <- tryCatch(predict(var_mod, n.ahead = h), error=function(e
    ) NULL)
399
400   if (!is.null(var_fc) && "y" %in% names(var_fc$fcst)) {
401     y_pred <- var_fc$fcst$y[, "fcst"]
402     metrics$VAR <- compute_metrics(ts_test, y_pred, insample = ts_
        train)
403
404     var_mod_full <- tryCatch(vars::VAR(mat, p = p_sel, type = "const"
        ), error=function(e) NULL)
405
406     var_fc_full <- tryCatch(predict(var_mod_full, n.ahead = h), error
        =function(e) NULL)
407
408     if (!is.null(var_fc_full) && "y" %in% names(var_fc_full$fcst)) {
409       ypf <- var_fc_full$fcst$y[, "fcst"]
410       low <- var_fc_full$fcst$y[, "lower"]
411       upp <- var_fc_full$fcst$y[, "upper"]
412       forecasts_full$VAR <- list(mean = ts(ypf, frequency = 12),
          lower = cbind(`80%`=low, `95%`=low)

```

```

405         upper = cbind(`80%`=upp, `95%`=upp))
406     }
407 }
408 }
409 }
410
411 # ---- Tabla de métricas unificada + mejor modelo ----
412 metr_df <- do.call(rbind, lapply(names(metricas), function(nm) cbind(
413     Modelo = nm, metricas[[nm]])))
414
415 rownames(metr_df) <- NULL
416
417 metr_df <- metr_df %>% dplyr::mutate(across(-Modelo, ~round(., 4))) %>%
418     dplyr::relocate(Modelo)
419
420 mejor <- metr_df$Modelo[which.min(metr_df$RMSE)]
421
422 updateSelectInput(session, "modelo_manual", choices = metr_df$Modelo,
423     selected = mejor)
424
425 list(
426     df = df,
427     y = y,
428     h = h,
429     cov_df = cov_df,
430     futuro_cov = futuro_cov,
431     forecasts = forecasts_full,
432     metrics = metr_df,
433     mejor = mejor,

```

```

427     var_dep = input$var_dep
428   )
429 })
430
431 # -- 4.6 Tabla comparativa de modelos --
432 output$tabla_modelos <- DT::renderDT({
433   req(resultados())
434   DT::datatable(resultados()$metrics, rownames = FALSE,
435                 options = list(scrollX = TRUE, dom = 't',
436                               columnDefs = list(list(className='dt-
437                                                     center', targets="_all"))))
438 })
439
440 # -- 4.7 Gráfico Mejor Modelo --
441 output$plot_mejor <- renderPlotly({
442   req(resultados())
443   res <- resultados()
444   best <- res$mejor
445   fc <- res$forecasts[[best]]
446   validate(need(!is.null(fc), "No hay forecast disponible para el mejor
447     modelo."))
448   fechas_hist <- res$df$Fecha
449
450   dfp <- if (inherits(fc, "forecast")) df_forecast_to_plot(fc, fechas_
451     hist) else {

```

```

449     n_hist <- length(fechas_hist)
450     fechas_fut <- seq(from = max(fechas_hist) %m+% months(1), by = "month
        ", length.out = length(fc$mean))
451     data.frame(
452         Fecha      = c(fechas_hist, fechas_fut),
453         Valor      = c(rep(NA, n_hist), as.numeric(fc$mean)),
454         Inferior   = c(rep(NA, n_hist), as.numeric(fc$lower[,1])),
455         Superior   = c(rep(NA, n_hist), as.numeric(fc$upper[,1])),
456         Tipo       = c(rep("Histórico", n_hist), rep("Proyección", length(fc$
            mean))))
457     )
458 }
459
460 df_hist <- data.frame(Fecha = res$df$Fecha, Valor = res$y, Tipo = "
        Histórico")
461
462 p <- ggplot() +
463     geom_line(data = df_hist, aes(Fecha, Valor, color = "Histórico"),
        size = 1) +
464     geom_line(data = subset(dfp, Tipo=="Proyección"), aes(Fecha, Valor,
        color="Proyección"), size = 1) +
465     geom_ribbon(data = subset(dfp, Tipo=="Proyección"),
466         aes(x=Fecha, ymin=Inferior, ymax=Superior, fill="IC"),
            alpha = 0.2) +

```

```

467     scale_color_manual(values = c("Histórico"="black", "Proyección"="#3366
      CC")) +
468     scale_fill_manual(values = c("IC"="#3366CC")) +
469     labs(title = paste("Mejor Modelo:", best), x = "Fecha", y = res$var_
      dep) +
470     theme_minimal() +
471     theme(legend.title = element_blank())
472
473     ggplotly(p, tooltip = c("x", "y", "Inferior", "Superior"))%>%
474     layout(legend = list(title = list(text = "")))
475 })
476
477
478 # -- 4.8 Gráfico: Todos los modelos --
479 output$plot_todos <- renderPlotly({
480     req(resultados())
481     res <- resultados()
482     fechas_hist <- res$df$Fecha
483     df_hist <- data.frame(Fecha = fechas_hist, Valor = res$y, Modelo = "
      Histórico")
484
485     fc_to_df <- function(fc, label) {
486         if (is.null(fc)) return(NULL)
487         n <- length(fc$mean)

```

```

488     fechas_fut <- seq(from = max(fechas_hist) %m+% months(1), by = "month
      ", length.out = n)
489     data.frame(Fecha = fechas_fut, Valor = as.numeric(fc$mean), Modelo =
      label)
490   }
491
492   dfl <- list(
493     fc_to_df(res$forecasts$ETS, "ETS"),
494     fc_to_df(res$forecasts$ARIMA, "ARIMA"),
495     fc_to_df(res$forecasts$SARIMA, "SARIMA"),
496     fc_to_df(res$forecasts$ARIMAX, "ARIMAX"),
497     fc_to_df(res$forecasts$SARIMAX, "SARIMAX"),
498     fc_to_df(res$forecasts$PROPHET, "PROPHET"),
499     fc_to_df(res$forecasts$VAR, "VAR")
500   )
501   df_all <- do.call(rbind, Filter(Negate(is.null), dfl))
502
503   p <- ggplot() +
504     geom_line(data = df_hist, aes(Fecha, Valor, color = "Histórico"),
      size = 1) +
505     geom_line(data = df_all, aes(Fecha, Valor, color = Modelo), size =
      1) +
506     labs(title = "Comparación de Pronósticos", x = "Fecha", y = res$var_
      dep) +
507     theme_minimal() +

```

```

508     scale_color_manual(name = "Modelos", values = c(
509         "Histórico" = "black",
510         "ETS" = "blue",
511         "ARIMA" = "red",
512         "SARIMA" = "green",
513         "ARIMAX" = "purple",
514         "SARIMAX" = "orange",
515         "PROPHET" = "brown",
516         "VAR" = "pink"
517     ))
518
519     ggplotly(p, tooltip = c("x", "y", "color"))%>%
520         layout(legend = list(title = list(text = "Modelos")))
521 })
522
523
524 # -- 4.9 Proyección de Covariables (subplots) --
525 output$plot_covs <- renderPlotly({
526     req(resultados())
527     res <- resultados()
528     cov_df <- res$cov_df
529     fut <- res$futuro_cov
530     validate(need(!is.null(cov_df) && ncol(cov_df) >= 1, "No hay
531         covariables seleccionadas."))

```

```

532 fechas_hist <- res$df$Fecha
533 fechas_fut  <- seq(from = max(fechas_hist) %m+% months(1), by = "month"
    , length.out = res$h)
534
535 plots <- lapply(seq_along(colnames(cov_df)), function(i) {
536   nm <- colnames(cov_df)[i]
537
538   df_h <- data.frame(Fecha = fechas_hist, Valor = as.numeric(cov_df[[nm
    ]]), Tipo = "Histórico")
539   df_p <- if (!is.null(fut) && nm %in% colnames(fut)) {
540     data.frame(Fecha = fechas_fut, Valor = as.numeric(fut[, nm]), Tipo
    = "Proyección")
541   } else {
542     data.frame(Fecha = fechas_fut, Valor = NA_real_, Tipo = "Proyección
    ")
543   }
544
545   df_plot <- rbind(df_h, df_p)
546
547   gg <- ggplot(df_plot, aes(x = Fecha, y = Valor, color = Tipo)) +
548     geom_line(size = 1) +
549     scale_color_manual(
550       name = "Tipo",
551       values = c("Histórico" = "black", "Proyección" = "#3366CC")
552     ) +

```

```

553     labs(x = NULL, y = nm) +
554     theme_minimal() +
555     theme(
556         axis.title.x = element_blank(),
557         axis.title.y = element_text(size = 10, color = "black") # <--
558         tamaño, negrita, color
559     )
560
561     # Mostrar leyenda solo en el primer gráfico
562     ggplotly(gg, tooltip = c("x", "y", "color")) %>%
563     layout(showlegend = ifelse(i == 1, TRUE, FALSE))
564 })
565
566 # Combinar subplots y ajustar layout para título y leyenda
567 subplot(
568     plots,
569     nrows = ceiling(length(plots)/2),
570     shareX = FALSE,
571     shareY = FALSE,
572     margin = 0.15,
573     titleX = TRUE,
574     titleY = TRUE
575 ) %>%
576 layout(

```

```

576     title = list(text = "Proyecciones de Covariables", x = 0.5, xanchor
                    = "center"),
577     legend = list(title = list(text = "Tipo"), orientation = "h", x =
                    0.5, xanchor = "center", y = 1.05),
578     margin = list(t = 60) # <-- más espacio arriba para el título
579 )
580 })
581
582
583
584 # -- 4.10 Proyección Conjunta (barras cacao + líneas covariables) --
585 output$plot_conjunto <- renderPlotly({
586     req(resultados())
587     res <- resultados()
588     best <- res$mejor
589     fc <- res$forecasts[[best]]
590     validate(need(!is.null(fc), "No hay forecast disponible para el mejor
                    modelo."))
591
592     n <- length(fc$mean)
593     fechas_fut <- seq(from = max(res$df$Fecha) %m+% months(1), by = "month"
                    , length.out = n)
594     cacao <- as.numeric(fc$mean)
595     low <- as.numeric(fc$lower[,1])
596     upp <- as.numeric(fc$upper[,1])

```

```

597
598 fut <- res$futuro_cov
599
600 # Construir dfp base con cacao y su intervalo
601 dfp <- data.frame(
602   Fecha = fechas_fut,
603   Cacao = cacao,
604   IC_Lower = low,
605   IC_Upper = upp
606 )
607
608 # Agregar dinámicamente todas las covariables del futuro
609 if (!is.null(fut) && ncol(fut) > 0) {
610   for (cov in colnames(fut)) {
611     dfp[[cov]] <- as.numeric(fut[, cov])
612   }
613 }
614
615 # Transformar a formato largo para ggplot
616 cov_long <- dfp %>%
617   pivot_longer(
618     cols = setdiff(colnames(dfp), c("Fecha", "Cacao", "IC_Lower", "IC_
619       Upper")),
620     names_to = "Covariable",
621     values_to = "Valor"

```

```

621     ) %>%
622     filter(!is.na(Valor))
623
624     # Crear gráfico
625     g <- ggplot() +
626     geom_col(data = dfp, aes(x = Fecha, y = Cacao), fill = "#8B4513",
627             alpha = 0.8) +
628     geom_ribbon(data = dfp, aes(x = Fecha, ymin = IC_Lower, ymax = IC_
629             Upper), fill = "#8B4513", alpha = 0.2) +
630     geom_line(data = cov_long, aes(x = Fecha, y = Valor, color =
631             Covariable), size = 1) +
632
633     labs(title = paste("Proyección conjunta (Mejor modelo:", best, ")"),
634           y = "Valores proyectados", x = "Fecha") +
635     theme_minimal()
636
637
638
639     # Convertir a plotly
640     ggplotly(g, tooltip = c("x", "y", "ymin", "ymax", "color")) %>%
641     layout(legend = list(title = list(text = "Covariables")))
642
643 })
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

643 req(resultados())
644 res <- resultados()
645 h <- res$h
646 fechas_fut <- seq(from = max(res$df$Fecha) %m+% months(1), by = "
        month", length.out = h)
647
648 get_mean <- function(fc) {
649     if (is.null(fc)) return(rep(NA_real_, h))
650     as.numeric(fc$mean)
651 }
652
653 out <- data.frame(
654     Fecha = fechas_fut,
655     ETS = get_mean(res$forecasts$ETS),
656     ARIMA = get_mean(res$forecasts$ARIMA),
657     SARIMA = get_mean(res$forecasts$SARIMA),
658     ARIMAX = get_mean(res$forecasts$ARIMAX),
659     SARIMAX = get_mean(res$forecasts$SARIMAX),
660     PROPHET = get_mean(res$forecasts$PROPHET),
661     VAR = get_mean(res$forecasts$VAR)
662 )
663 writexl::write_xlsx(out, file)
664 }
665 )
666

```

```
667  
668 }  
669  
670 # ---- 5) Lanzar App ----  
671 shinyApp(ui = ui, server = server)
```

Código A.1. Código de la aplicación Shiny (R) para el pronóstico de producción de cacao.