

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA
ELABORACION DE PRONOSTICOS DE PRODUCCION, EMPLEANDO SERIES
DE TIEMPO**

DAVID ESNEYDER BELLO ANGULO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-QUIMICAS
ESCUELA DE INGENIERÍA DE PETRÓLEOS
BUCARAMANGA**

2017

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA
ELABORACION DE PRONOSTICOS DE PRODUCCION, EMPLEANDO SERIES
DE TIEMPO**

DAVID ESNEYDER BELLO ANGULO

Trabajo de grado para optar al título de Ingeniero de Petróleos.

Director

ERIK GIOVANY MONTES PAEZ

Ingeniero de Petróleos, M.Sc.

Co-director

CARLOS ALFONSO MANTILLA DUARTE

Economista, Especialista en estadística.

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-QUIMICAS
ESCUELA DE INGENIERÍA DE PETRÓLEOS
BUCARAMANGA**

2017

"El caos es un orden por descifrar"

José Saramago

El hombre duplicado

AGRADECIMIENTOS

Al que vive y reina por los siglos de los siglos. “Señor, digno eres de recibir la gloria y la honra y el poder; porque tú creaste todas las cosas, y por tu voluntad existen y fueron creadas” (Apocalipsis 4: 11, RVR1960).

A Lizeth Angulo Ardila por su incansable lucha para traerme hasta aquí, gracias por lograrlo, me enorgullece tener el privilegio de llamarte mamá. El ejemplo que me has dado es la mejor enseñanza que jamás recibiré

A mi “Mamita”, Delia Mercedes Ardila Galeano, porque con amor y dedicación me cuidó por casi dos décadas, me dio cariño y me enseñó a valorar las cosas. Es maravilloso que la madre de mi madre sea una persona tan ejemplar.

A Erley Angulo Ardila por estar siempre dispuesta a escucharme, por ayudarme, animarme, y, sobre todo, por no perder la fe y no haberse dado por vencida hasta que entregué mi vida a Dios. Lo digo y lo seguiré diciendo, tengo tres madres.

A Walter Ebed Angulo Ardila por ser esa figura paterna a quien deseo parecerme, por enseñarme que el trabajo es parte vital del ser humano y que se debe dar lo máximo para lograr las metas.

A Valentina Argüellez Angulo porque desde que nació veo por sus ojos y me ha mostrado la parte tierna de la vida.

A mi amor Natalia Ivonne Suárez Barajas por su apoyo incondicional, por hacerme reír aun cuando estoy triste, por hacer que cada 19 sea especial y por estos dos años de gran felicidad, anhelo que sean muchos más.

A mi director, Erik Giovany Montes Páez por todo el tiempo, el apoyo, la paciencia y el interés mostrado en el desarrollo de este software, porque en vez de simplemente esperar informes en fechas planteadas respondió mis inquietudes casi a diario. Por ser un buen cristiano y un buen amigo, gracias.

Al profesor Carlos Alfonso Mantilla Duarte, por sus charlas sobre estadística, por generar curiosidad e incentivar la búsqueda del conocimiento, por la ayuda que me brindó incluso en época decembrina, por la iniciativa de enviarme información útil, por hablar como un amigo, gracias.

A los desarrolladores de R, por realizar un software libre de tal calidad y de esta manera brindar tantas posibilidades a todo aquel que desee explorar su potencial.

A Microsoft por ofrecer un IDE muy cómodo e intuitivo que facilita en gran manera el desarrollo de aplicaciones.

A todos aquellos que con sus aportes a la ciencia y sus trabajos de investigación hicieron posible el desarrollo de este proyecto.

A los calificadores, el ingeniero Fernando Enrique Calvete y el ingeniero Edinson Odilio Garcia por saber valorar el esfuerzo realizado.

A todos mis familiares y amigos que me dieron ánimo y me apoyaron para el cumplimiento de este objetivo.

A todos aquellos que me ayudaron a cumplir este sueño respetando mis creencias sobre el día de reposo y adoración a Dios.

A los escritores que me acompañaron con sus obras en mis tiempos libres, en especial a José Saramago, Q.E.P.D., por sus escritos, que plantean cuestiones tan grandes y profundas.

A los desarrolladores de videojuegos que ponen todo su esfuerzo en grandes proyectos cuyo propósito es divertir y recrear a las personas.

David Esneider Bello Angulo

TABLA DE CONTENIDO

INTRODUCCIÓN	18
1.CONCEPTOS FUNDAMENTALES.....	20
1.1.DECLINACION DE LA PRODUCCIÓN.....	20
1.1.1.Factores que afectan las curvas de declinación	20
1.1.2.Tipos de declinación	22
1.1.3.Antecedentes	25
1.1.4.Herramientas software utilizadas en pronósticos de producción	30
1.2.SERIES DE TIEMPO	32
1.2.1.Componentes de una serie de tiempo	32
1.2.2.Métodos de pronóstico.....	33
1.3.SERIES ESTACIONARIAS EN DIFERENCIAS Y EN TENDENCIAS.....	35
1.4.METODOLOGIA DE BOX & JENKINS (ARIMA).....	38
1.4.1.Identificación tentativa:.....	40
1.4.2.Estimación de los coeficientes del modelo:.....	44
1.4.3.Comprobación y diagnóstico.....	44
1.4.4.Pronóstico	46
2.HERRAMIENTAS EMPLEADAS PARA EL DESARROLLO Y FUNCIONAMIENTO INTERNO DEL SOFTWARE.....	46
2.1.VISUAL STUDIO COMMUNITY 2015.....	47
2.2.R.....	48
2.3.R.NET	50

2.3.1.Funciones principales	52
2.4.PAQUETE DE INTEROPERABILIDAD ENTRE MICROSOFT EXCEL Y VISUAL STUDIO	53
2.5.FUNCIONAMIENTO INTERNO DEL SOFTWARE	56
2.5.1.Carga de datos	56
2.5.2.Modelos usados.....	57
2.5.3.Realización del pronóstico	64
3.MANUAL DE USUARIO.....	66
3.1.INSTALACIÓN Y EJECUCIÓN DEL SOFTWARE.....	68
3.1.1.Instalación.....	68
3.1.2.Ejecutar DeclineR 1.0 por primera vez.....	71
3.2.CARGA DE DATOS	75
3.2.1.Copia de datos desde Excel	75
3.2.2.Carga directa de archivos	78
3.3.MANEJO DE DATOS.....	82
3.3.1.Ajuste de los datos.....	83
3.3.2.Selección de datos.....	83
3.4.CONSTRUCCIÓN Y ELECCIÓN DEL MODELO.....	86
3.4.1.Construcción de un modelo Exponencial (Arps)	87
3.4.2.Construcción de un modelo Armónico (Arps).....	88
3.4.3.Construcción de un modelo Hiperbólico (Arps).....	90
3.4.4.Construcción de un modelo ARIMA (Box & Jenkins)	95
3.4.5.Modelo combinado.....	99
3.4.6.Resultados del modelo.....	99

3.5.PRONÓSTICO.....	101
3.5.1.Generar pronóstico	101
3.5.2.Generar informe.....	101
4.EJEMPLOS DE APLICACIÓN Y ANÁLISIS COMPARATIVO DEL ERROR.....	110
4.1.ANALISIS COMPARATIVO DEL ERROR A CORTÍSIMO PLAZO	111
4.1.1.Construcción de los modelos	111
4.1.2.Resultados obtenidos.....	113
4.2.ANÁLISIS DEL ERROR OBTENIDO A MEDIANO PLAZO.....	117
4.2.1.Caso 1.....	118
4.2.2.Caso 2:.....	121
5.CONCLUSIONES	125
6.RECOMENDACIONES	127
BIBLIOGRAFIA.....	128
ANEXOS	131

LISTA DE FIGURAS

Figura 1.1.1: Declinación Exponencial.....	22
Figura 1.1.2: Declinación Hiperbólica	23
Figura 1.1.3: Declinación Armónica.	24
Figura 1.1.4: Comparación de los tipos de declinación. a) Log(q) vs. Tiempo, b) Log(q) vs. Neto Producido, c) Producción vs. Tiempo	25
Figura 1.3.1: Tendencia estocástica y determinista.	37
Figura 1.3.2: Función de autocorrelación simple.	41
Figura 1.3.3: Función de autocorrelación parcial.	42
Figura 1.3.4: FAS y FAP típicas para determinados modelos.....	43
Figura 1.3.5: Valor P.	45
Figura 2.1.1: IDE Visual Studio Community.	49
Figura 2.2.1: Descarga de R.Net desde NuGet.	50
Figura 2.4.1: Paquete de interoperabilidad entre Excel y Visual Studio.....	53
Figura 2.5.1: Explicación del algoritmo para encontrar el mejor modelo.....	61
Figura 3.1.1. Asistente para la instalación de DeclineR 1.0.....	69
Figura 3.1.2: elección de la carpeta de instalación.	70
Figura 3.1.3: Pantalla de inicio:.....	71
Figura 3.1.4: Opción de instalación automática de R.....	72
Figura 3.1.5: Descarga interna de R.	73
Figura 3.1.6: Asistente para la instalación de R.....	74
Figura 3.2.1: Ruta para copiar datos.....	75
Figura 3.2.2: Formulario para copiar datos.	76
Figura 3.2.3: Selección de fechas.....	77
Figura 3.2.4: Formulario para cargar archivos.	78
Figura 3.2.5: Abrir un archivo de datos.	79
Figura 3.2.6: Cuadro de diálogo “Buscar Archivo”.	80

Figura 3.2.7: Primera vista del archivo cargado.....	81
Figura 3.2.8: Datos organizados y listos para cargar.....	82
Figura 3.3.2: Página de manejo de datos	85
Figura 3.4.1 Opciones para la construcción de un modelo exponencial.	88
Figura 3.4.2: Opciones para la construcción de un modelo armónico	89
Figura 3.4.3: Controles para un modelo hiperbólico de Arps.	89
Figura 3.4.5: Modelo Hiperbólico especificando Di y n (opción totalmente manual).	92
Figura 3.4.6: Modelo Hiperbólico especificando n (opción semiautomática).....	93
Figura 3.4.7: Modelo Hiperbólico automático.....	94
Figura 3.4.8: Función de Autocorrelación Simple.	96
Figura 3.4.9: Función de Autocorrelación Parcial.	97
Figura 3.4.10: Modelo ARIMA realizado con función auto.arima().....	98
Figura 3.4.11: Panel de resultados del modelo (Resultados de un modelo ARIMA).	100
Figura 3.4.12: Información de los modelos usados.....	100
Figura 3.5.1: Opción para definir qué datos irán en el informe.	103
Figura 3.5.2: Gráfico de un modelo ARIMA que incluye los datos del modelo previos al pronóstico.....	104
Figura 3.5.3: Gráfico de un modelo ARIMA que no incluye los datos del modelo previos al pronóstico.	104
Figura 3.5.4: Página “Pronóstico”.	105
Figura 3.5.5: Generar informe con el pronóstico únicamente (quince días pronosticados).	106
Figura 3.5.6: Generar informe incluyendo todos los datos que fueron usados (Treinta días pronosticados).....	107
Figura 3.5.7: Generar informe incluyendo los últimos treinta datos que fueron usados (treinta días pronosticados).....	108
Figura 3.5.8: Ejemplo, informe generado pronosticando quince días e incluyendo los últimos quince datos utilizados.	109

Figura 4.1.1: Grafico comparativo de pronósticos para el pozo 1.....113
Figura 4.1.2: Grafico comparativo de pronósticos para el pozo 2.....114
Figura 4.1.3: Grafico comparativo de pronósticos para el pozo 3.....115
Figura 4.2.1: Pronósticos para el caso 1.....119
Figura 4.2.2: Pronósticos para el caso 2.....123

LISTA DE TABLAS

Tabla 1.1.1: Exponente n para distintos tipos de Yacimiento.....	23
Tabla 2.1.1: Objetos de R, R.Net y .Net relacionados según su tipo.	51
Tabla 2.5.1: Matriz “m” para la regresión por mínimos cuadrados.....	58
Tabla 2.5.2: Vector “y” para la regresión por mínimos cuadrados	59
Tabla 2.5.3: Ejemplo de una matriz de criterios de información de Akaike.....	63
Tabla 4.1.1: Resultados de DeclineR para los modelos del pozo 1	111
Tabla 4.1.2: Resultados de DeclineR para los modelos del pozo 2	112
Tabla 4.1.3: Resultados de DeclineR para los modelos del pozo 3.....	112
Tabla 4.1.4: Errores obtenidos por los modelos para todos los pozos.....	117
Tabla 4.2.1: Resultados de DeclineR para los modelos del Caso 1.	118
Tabla 4.2.2: Errores presentados por los pronósticos para el caso 1.	120
Tabla 4.2.2: Resultados de DeclineR para los modelos del Caso 2.	122
Tabla 4.2.2: Errores presentados por los pronósticos para el caso 1.	124

LISTA DE ANEXOS

Anexo A. Historiales de producción de los pozos utilizados.....	131
Anexo B. Código de la case “Constructor de modelos”.....	133

RESUMEN

TÍTULO: DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA ELABORACIÓN DE PRONÓSTICOS DE PRODUCCIÓN, EMPLEANDO SERIES DE TIEMPO*

AUTOR: DAVID ESNEYDER BELLO ANGULO**

PALABRAS CLAVE: Curvas de declinación, pronósticos, series de tiempo, R Project, Interoperabilidad entre R y Visual Studio, ARIMA, Curvas de Arps.

DESCRIPCIÓN:

En el año 2016 Montes Erick planteó una metodología para el modelado de los pronósticos de producción, tomando el historial de pozo como una serie de tiempo estacionaria en diferencias, para de esta manera aplicar el modelo ARIMA de Box y Jenkins con el fin de mejorar el ajuste de la predicción. En su trabajo obtuvo resultados favorables para el cortísimo plazo, con el inconveniente de que el procedimiento para realizar la metodología resulta muy complejo para el usuario promedio y requiere una alta disponibilidad de tiempo.

El presente trabajo se desarrolló con el objetivo de automatizar dicha metodología en un software que pueda ser utilizado por la industria. Además, se incluyeron otros métodos con el fin de que el usuario elija el más adecuado, entre los que se destacan el modelo hiperbólico de Arps con mejor coeficiente de determinación, y una variante a la metodología mencionada, tomando la serie como estacionaria en tendencia, para así determinar la componente tendencial con las curvas Arps, y las otras componentes con el modelo de Box y Jenkins.

Como alcance adicional se incluye una breve explicación sobre cómo utilizar el software R desde las aplicaciones realizadas en Visual Studio, utilizando el paquete R.Net, que permite ingresar valores, realizar operaciones y extraer resultados, aprovechando todas las ventajas que ofrece R para el análisis de datos y la amplia gama de librerías que tiene disponibles.

* Trabajo de grado.

** Facultad de Ingenierías Físicoquímicas. Escuela de Ingeniería de Petróleos. Director: Ing. Erik Giovany Montes Páez, M.Sc. Codirector: Carlos Alfonso Mantilla Duarte, Economista, Especialista en Estadística.

ABSTRACT

TITLE: DEVELOPMENT OF A SOFTWARE TOOL FOR THE ELABORATION OF PRODUCTION FORECASTS, USING TIME SERIES.*

AUTHOR: DAVID ESNEYDER BELLO ANGULO**

KEYWORDS: Decline curves, forecast, time series, R Project, interoperability between R and Visual Studio, ARIMA, Arps's curves.

DESCRIPTION:

In 2016 Montes Erik raised a methodology for the modeling of production forecasts, it takes the well history as a in difference stationary time series, to apply the Box and Jenkins ARIMA model to improve the production's fit. He got good results in his work for the very short term, but the process results very complicated to average user and it requires a long availability of time.

The present work was developed with the aim of automating this methodology in a software tool that can be used by the petroleum industry. Also, it includes other methods to user can choose the most suitable one, among which stands out Arps's hyperbolic model with the best coefficient of determination, and a variant of the mentioned methodology, taking the series as a trend stationary time series to determinate trend component with Arps's curves and the other ones with the model of Box and Jenkins.

As an additional scope, it includes a brief explication over how to use R software from applications developed with Visual Studio, using the R.Net package, which allows the input of values to do operations and to extract results, taking advantage of all options offered by R for data analysis, and the wide range of available libraries.

* Degree work.

** Physical-Chemical Faculty of Engineering. Petroleum Engineering School. Director: Eng. Erik Giovany Montes Páez, M.Sc. Codirector: Carlos Alfonso Mantilla Duarte, Economist, Statistics Specialist.

INTRODUCCIÓN

Después de que en el año 2015 Antolines B. y Carrillo R.¹ mostraran el amplio margen de error que manejan los modelos utilizados para elaborar pronósticos de producción, en el 2016 Montes Erik² planteó tomar la historia de producción como una serie de tiempo estacionaria en diferencias para aplicar la metodología ARIMA³ de Box y Jenkins como alternativa, obteniendo buenos resultados en pronósticos a cortísimo plazo, sin embargo, su metodología, que resulta ser compleja para el usuario común, implica una alta disponibilidad de tiempo para hacer un análisis y elegir el mejor orden para el modelo. Este trabajo inició con el objetivo de automatizar esa metodología y ofrecer un software que pudiera ser utilizado por la industria, objetivo que se cumplió, incluyendo además opciones para realizar pronósticos con las curvas empíricas de Arps, destacando el modelo hiperbólico que puede ser hallado automáticamente, incluso cuando se necesiten exponentes de declinación atípicos.

Los modelos más ampliamente utilizados para la elaboración de pronósticos de producción son las curvas de Arps⁴, propuestas en 1945, pero teniendo en cuenta que la historia de producción es una serie de tiempo, pueden utilizarse técnicas estadísticas como la de Box & Jenkins en la realización de pronósticos.

¹ ANTOLÍNEZ, Brayan. CARRILLO, Reynel. Estudio comparativo de los modelos utilizados en la generación de pronósticos de producción para yacimientos con fluidos composicionales. Trabajo de grado. Bucaramanga: Universidad Industrial de Santander, 2015.

² MONTES, Erick. Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos. Trabajo de grado (Magíster en Ingeniería de Hidrocarburos) Universidad Industrial de Santander. Bucaramanga. 2016.

³ G.P.E. Box y G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1978.

⁴ ARPS, J. Analysis of decline curves. The British-American Oil Producing Co. SPE 945228-G. 228-247.(1945).

Existen en la actualidad diversas herramientas gratuitas que permiten obtener productos útiles para la comunidad, tal es el caso de Visual Studio y R, que fueron utilizados para desarrollar la herramienta software DeclineR que permite realizar diversos modelos que describan el historial de producción de un pozo determinado y pronósticos a partir de dichos modelos, generando finalmente un informe.

Adicionalmente, se propuso una variante para la metodología planteada por Montes Erik, tomando la serie como estacionaria en tendencia para combinar de esta manera el modelo de Arps y el de Box & Jenkins, con la intención de mejorar los pronósticos a corto plazo.

Para corroborar la utilidad del software se realizó un análisis comparativo del error obtenido a cortísimo y mediano plazo por DeclineR y herramientas comerciales.

Los resultados obtenidos fueron favorables para los casos de estudio, pues mostraron una mejora significativa en los pronósticos a cortísimo plazo, y también en los realizados a uno y dos años. La metodología propuesta para combinar los modelos hiperbólico y ARIMA, mostró resultados favorables en los casos mencionados.

1. CONCEPTOS FUNDAMENTALES

1.1. DECLINACION DE LA PRODUCCIÓN

Es la disminución de la producción de crudo o gas de un pozo a través de su vida productiva a causa de factores como la disminución en la presión de yacimiento. Arps (1945)⁵ propuso curvas de declinación de tipo exponencial, hiperbólica y armónica, que representan la declinación.

1.1.1. Factores que afectan las curvas de declinación: Según Magdalena Paris de Ferrer⁶ existen muchos factores que afectan la declinación, entre los cuales se encuentran los siguientes:

- Periodos desiguales de tiempo:

Las pruebas de pozos y las mediciones no se efectúan en intervalos de tiempo regulares, esto afecta los promedios, de manera que lo ideal es que los lapsos de tiempo se mantengan constantes.

- Cambio de productividad de los pozos:

⁵ ARPS, J. Analysis of decline curves. The British-American Oil Producing Co. SPE 945228-G. 228-247.(1945).

⁶ PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009. (p. 505-518).

Debido a que la productividad de los pozos disminuye como efecto de la declinación, algunos pozos son estimulados mediante distintas técnicas, lo cual genera un aumento en la producción y por ende un cambio en la tendencia de declinación.

- Terminación de nuevos pozos:

Esto afecta la declinación a nivel de yacimiento, ya que el nuevo pozo aporta producción, y puede cambiar la tendencia.

- Interrupción de los programas de producción:

Los cierres de los pozos son una fuente importante de ruido en la declinación del pozo, ya que reduce el caudal a cero en un periodo de tiempo. Si no se trata adecuadamente, puede tener un impacto negativo sobresaliente en las curvas de tendencia.

- Veracidad de datos:

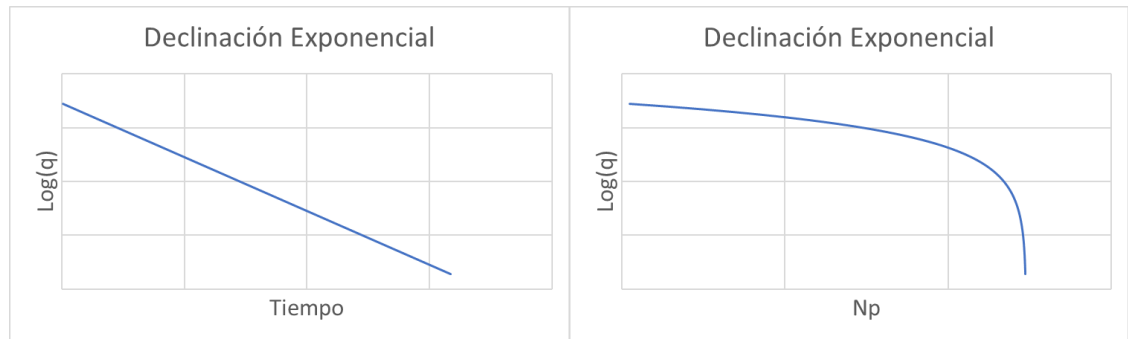
En campos antiguos especialmente, se desconoce la calidad de la información suministrada, y si los datos no son verdaderos, generan gran error al analizar curvas de declinación. Por este motivo se prefiere trabajar con datos recientes, que suelen ser más confiables.

1.1.1.1. Tipos de declinación⁷: La clasificación de la declinación normalmente se hace con respecto al comportamiento de la producción, y está directamente relacionada con las curvas de Arps. Con base en esto, existen tres tipos de declinación:

1.1.1.2. Declinación exponencial: Se mantiene un porcentaje o tasa de declinación constante.

$$q_t = q_i * e^{-D_i * t} \quad (\text{Ec.1.1.1})$$

Figura 1.1.1: Declinación Exponencial



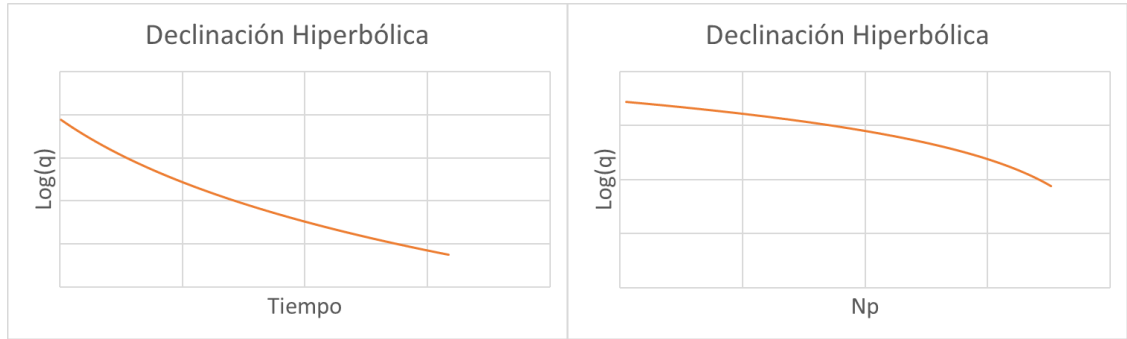
Fuente: PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009.

1.1.1.3. Declinación hiperbólica: “La caída de producción por unidad de tiempo como una fracción de la tasa de producción es proporcional a una potencia fraccional de la tasa de producción, con la potencia variando entre 0 y 1”

⁷ PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009.

$$q_t = q_i * (1 + n * Di * t)^{-1/n} \quad (\text{Ec. 1.1.2})$$

Figura 1.1.2: Declinación Hiperbólica



Fuente: PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009.

El exponente de declinación “n” varía dependiendo del tipo de yacimiento y de empuje (ver Tabla 1.1).

Tabla 1.1.1: Exponente n para distintos tipos de Yacimiento

CASO	N
Empuje por gas en solución	Cerca de 0,3
Pozos de gas	0,4 a 0,5
Líquido una sola fase	Cerca de 0
Yacimiento de petróleo con empuje de agua	Cerca de 0,5

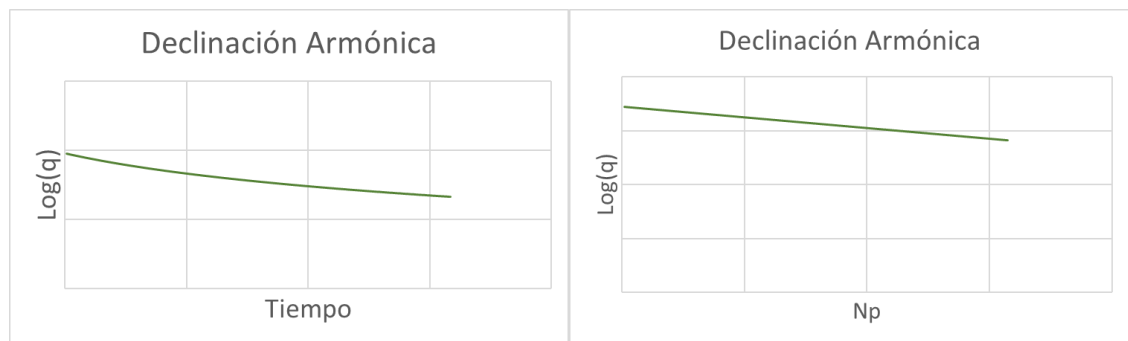
Fuente: FETKOVICH, M.J., FETKOVICH, E.J., & FETKOVICH, M.D. Useful Concepts for Decline-Curve Forecasting, Reserve Estimation, and Analysis. Paper SPE 28628. 1994

1.1.1.4. Declinación armónica:

$$q_t = q_i * (1 + n * Di * t)^{-1} \quad (\text{Ec. 1.1.3})$$

La caída en producción por unidad de tiempo como una fracción de la tasa de producción es directamente proporcional a la tasa de producción.

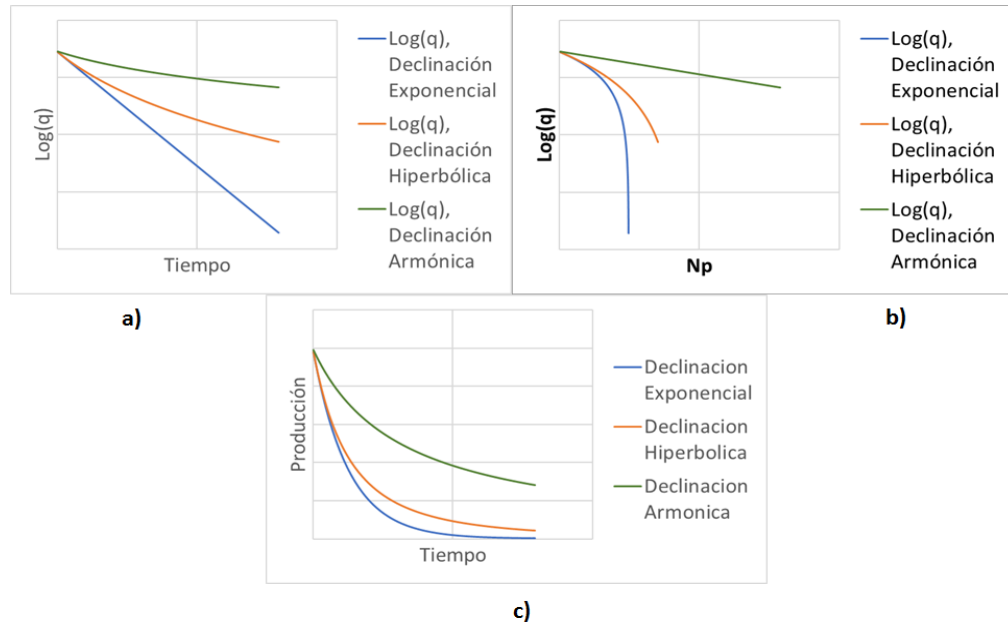
Figura 1.1.3: Declinación Armónica.



Fuente: PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009.

Al comparar los 3 tipos de declinación, se puede encontrar que el agotamiento mas temprano ocurre con la declinación exponencial, seguido por la hiperbólica y la armónica respectivamente (Ver Figura 1.4).

Figura 1.1.4: Comparación de los tipos de declinación. a) $\text{Log}(q)$ vs. Tiempo, b) $\text{Log}(q)$ vs. Neto Producido, c) Producción vs. Tiempo



1.1.2. Antecedentes⁸: Entre los principales aportes al estudio de curvas de declinación se pueden encontrar los siguientes:

⁸ ANTOLÍNEZ, Brayan. CARRILLO, Reynel. ESTUDIO COMPARATIVO DE LOS MODELOS UTILIZADOS EN LA GENERACIÓN DE PRONÓSTICOS DE PRODUCCIÓN PARA YACIMIENTOS CON FLUIDOS COMPOSICIONALES. Trabajo de grado. Bucaramanga: Universidad Industrial de Santander, 2015.

- 1908, Arnold y Anderson⁹ fueron los primeros en hablar del termino declinación, y la definieron como una fracción de caída de presión aproximadamente constante.
- 1924, Cutler¹⁰ propuso una relación hiperbólica en papel logarítmico para representar mejor el comportamiento de la declinación, señalando que los modelos geométricos y exponenciales eran muy conservadores en cuanto al neto producido.
- 1944, Arps¹¹ presentó un estudio sobre el análisis de las curvas de declinación, el cual aunque originalmente se consideró muy empírico, pero fue muy aceptado gracias a su simplicidad y efectividad en los pronósticos.
- Gentry¹², en 1972, presentó un método grafico para el análisis de los tres tipos de declinación, con base en el tiempo, y en el neto producido.
- 1973, Fetkovich¹³, mejoró una idea presentada por Slider¹⁴ en 1968 combinando la solución para presión de fondo constante, con la solución de Arps, presentando una curva tipo.

⁹ ARNOLD, R. y ANDERSON, R. Preliminary Report on Coalinga Oil District. U.S. Geological Survey Bulletin, 357 (1908) 79.

¹⁰ CUTLER, W.W. Jr. Estimation of Underground Oil Reserves by Well Production Curves. U.S. Bureau of Mines Bulletin, 228 (1924).

¹¹ ARPS, J.J. Analysis of Decline Curves. Trans., AIME (1945) 160, 228-247.

¹² GENTRY, R.W. Decline Curve Analysis. JPT (January, 1972) 38-41.

¹³ FETKOVICH, M.J. Decline Curve Analysis Using Type Curves. JPT (June, 1980) 1065-1077

¹⁴ SLIDER, H.C. A Simplified Method of Hyperbolic Decline Curve Analysis. JPT (March, 1968) 235-236.

- 1978, Gentry y McCray¹⁵ estudiaron los efectos de las propiedades del fluidos y de la roca en la declinación de la producción, y concluyeron que la heterogeneidad del yacimiento aumenta los valores del exponente de declinación y puede hacer que sea mayor que 1. Además presentaron un método mejorado.
- 1986, Blasingame y Lee¹⁶ presentaron técnicas de análisis para una sola fase líquida en estado post-transitorio a presión de fondo constante. Las técnicas de análisis son dadas para pozos en yacimientos homogéneos, naturalmente fracturados y verticalmente fracturados. Un aporte importante fue el desarrollo de una relación para la presión promedio de yacimiento durante el flujo estabilizado para un escenario de tasa variable.
- 1987, Harrington¹⁷ introdujo un método computacional para pronosticar el rendimiento de pozos de gas y el cálculo de reservas de gas.
- 1989, Doung¹⁸ propuso un enfoque numérico para analizar curvas de declinación manipulando las ecuaciones empíricas de Arps para obtener una relación que es una combinación lineal de las variables q , N_p , y q_t , a la que

¹⁵ GENTRY, R.W. y McCRAY, A.W. The Effect of Reservoir and Fluid Properties on Production Decline Curves. JPT (September, 1978) 1327-1341.

¹⁶ BLASINGAME, T.A. y LEE, W.J.: Properties of Homogeneous Reservoirs, Naturally Fractured Reservoirs, and Hydraulically Fractured Reservoirs from Decline Curve Analysis, paper SPE 15018 presented at the 1986 SPE Permian basin Oil & Gas Recovery Conference, Midland, March 13-14.

¹⁷ HARRINGTON, A.G., et al: Application of Type-Curve Techniques to Decline Analysis and Forecasting of Gas Wells, paper SPE 16936 presented at the 1987 Annual Technical Conference and Exhibition of the SPE, Dallas, TX, September 27-30

¹⁸ DUONG, A.N. A New Approach for Decline-Curve Analysis. Paper SPE 18859 presented at the 1989 SPE Production Operations Symposium, Oklahoma, March 13-14

se le aplica una regresión lineal múltiple con el fin de encontrar los coeficientes óptimos para la ecuación hiperbólica de Arps.

- 1991, Blasingame, McCray y Lee¹⁹ basados en la presentación original de McCray²⁰, analizaron la declinación en condiciones de presión de fondo fluyente variable e introdujeron una función de tiempo que fue llamada “tiempo análogo de presión constante”, con el fin de transformar este problema en un sistema equivalente de presión de fondo constante,
- 1998, R.G. Agarwal, Gardner, Kleinsteiber, y Fussell,²¹ utilizando datos de producción, propusieron combinar los conceptos de curvas de declinación y curvas tipo en un método para la estimación de reservas y características del yacimiento, aplicable a pozos de gas y de aceite.
- Ese mismo año, L.O Masoner²² realizó un importante trabajo al deducir a partir de principios de permeabilidad relativa la ecuación hiperbólica de Arps, lo que mostró que esta declinación aplica en procesos de IOR y EOR dominados por la permeabilidad relativa.

¹⁹ BLASINGAME, T.A., et al. "Decline Curve Analysis for Variable Pressure Drop/Variable Flowrate Systems," paper SPE 21513 presented at the 1991 SPE Gas Technology Symposium, Houston.

²⁰ McCRAY, T.C. Reservoir Analysis Using Production Decline Data and Adjusted Time. M.S. Thesis, Texas A&M U., College Station, TX, 1990.

²¹ AGARWAL, R.G., et al. Analyzing well production data using combined type-curve and decline-curve analysis concepts. Paper SPE 49222, 1998.

²² MASONER, L.O. Decline analysis relationship to relative permeability in secondary and tertiary recovery. Paper SPE 39928, April 1998.

- 2001, F. Abdelhafidh y T. Djebbar²³ propusieron una ecuación de ajuste universal como alternativa al ajuste manual con curvas tipo, que combina las ecuaciones empíricas de Arps y la solución analítica de Fetkovich. Se presentaron ventajas como mayor precisión al calcular radio adimensional, exponente de declinación y puntos de partida, además, la data de producción puede ser montada sin usar técnicas de suavizado, las regiones de estado estable y pseudoestable son determinadas automáticamente, y algunos datos de producción insuficientes para ser analizados con curva tipo, pueden ser analizados con la ecuación de ajuste.
- 2012, Brito, L. E. Paz, F. Belisario, D. R²⁴ propusieron un enfoque estadístico, dividiendo cada tasa de producción en el tiempo por la inicial, obteniendo así un perfil de los factores con valores entre 0 y 1, que representa con mayor precisión el comportamiento de la declinación. Teniendo un grupo de pozos, plantearon crear envolventes de estas curvas, para luego conforme a un estudio estadístico pronosticar la producción de un pozo.
- En 2016 Montes, E. propuso el manejo de los datos de producción como una serie de tiempo, aplicando modelos estadísticos, como la metodología de Box & Jenkins (ARIMA), y métodos de estadística no paramétrica.

Este recuento histórico se puede encontrar ampliado en el trabajo realizado en el 2015 por Atolinez Brayan y Carrillo Reynel “Estudio comparativo de los modelos

²³ ABDELHAFIDH, F. y DJEBBAR, T. Application of decline-curve analysis technique in oil reservoir using a universal fitting equation. Paper SPE 70036, May 2001.

²⁴ BRITO, L. E., et al. Probabilistic Production Forecasts Using Decline Envelopes. Paper SPE 152392-MS, 2012.

utilizados en la generación de pronósticos de producción para yacimientos con fluidos composicionales.”²⁵

1.1.3. Herramientas software utilizadas en pronósticos de producción: En el año 2015 Carrillo R. y Antolínez B.²⁶ realizaron en su trabajo de grado un estudio comparativo de diferentes herramientas software empleadas para la elaboración de pronósticos de producción, que fue ampliado por Montes E.²⁷ en 2016. A continuación, se presentaran las herramientas que analizaron, con una breve descripción y los métodos que emplea cada una para la generación de pronósticos:

1.1.3.1. Oil Field Manager (OFM)²⁸: OFM de Schlumberger es un software de análisis de pozo y yacimiento que busca facilitar el acceso rápido a la información importante proporcionando datos útiles para administradores, ingenieros, o geólogos.

OFM utiliza las curvas empíricas de Arps para realizar los pronósticos, pero el usuario debe elegir la curva más apropiada.

1.1.3.2. Value Navigator (VAL NAV)²⁹: Valor Navigator® (Val NAV) de la empresa Energy Navigator es una herramienta que además de realizar predicciones, las evalúa económicamente para establecer el valor de las reservas.

²⁵ ANTOLÍNEZ, Brayan. CARRILLO, Reynel. ESTUDIO COMPARATIVO DE LOS MODELOS UTILIZADOS EN LA GENERACIÓN DE PRONÓSTICOS DE PRODUCCIÓN PARA YACIMIENTOS CON FLUIDOS COMPOSICIONALES. Trabajo de grado. Bucaramanga: Universidad Industrial de Santander, 2015.

²⁶ Ibíd.

²⁷ MONTES, Erick. Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos. Trabajo de grado (Magíster en Ingeniería de Hidrocarburos) Universidad Industrial de Santander. Bucaramanga. 2016.

²⁸ SCHLUMBERGER LIMITED. Oil Field Manager Well and Reservoir Analysis Software, 2009. Schlumberger 2016. ALL RIGHTS RESERVED.

²⁹ ENERGY NAVIGATOR. Value Navigator®, Energy Navigator 2016. ALL RIGHTS RESERVED.

En la generación de pronósticos utiliza también las curvas de Arps, pero elige la mejor, permitiendo tener modelos “súper- hiperbólicos” con exponentes mayores a 1, además de realizar una selección de los datos correctos y descartar datos erróneos automáticamente.

1.1.3.3. ECRIN V4.3.0 (Topaze NL)³⁰: Ecrin es un conjunto integrado de ingeniería con distintos módulos que incluyen el análisis de presión transitoria (Saphir NL), el análisis de la producción (Topaze NL), ajuste histórico Fullfield (Rubis) y el análisis de rendimiento del pozo (Amatista).

Topaze NL (usado para analizar el comportamiento de la producción) permite pronosticar la producción utilizando los modelos de Arps, Fetkovich o Blasingame, pero para esto requiere una gran cantidad de información como datos PVT, presiones a las que va a estar operando el pozo en el intervalo de tiempo que se desea pronosticar, entre otros.

1.1.3.4. FORECAST³¹: El software Forecast de RYDER SCOTT utiliza también las curvas de Arps para la generación de pronósticos, proporcionando un método automático, donde se elige automáticamente el tipo de declinación y se realiza el modelo con todos los datos, y otro donde el usuario puede realizar un ajuste manual, dividiendo los datos en múltiples segmentos en caso de ser necesario.

³⁰ KAPPA. Petroleum Exploration & Production Software Training & Consulting. Topaze NL, V4.2. Kappa: 1987-2016.

³¹ RYDER SCOTT COMPANY, L.P. Forecast Multi-Trend Decline Analysis. Ryder Scott Company, L.P: 2016.

1.2. SERIES DE TIEMPO

“Una serie de tiempo es un conjunto de observaciones de una variable medida en puntos sucesivos en el tiempo o en periodos de tiempo sucesivos.”³². El análisis de las series de tiempo tiene como objetivo la predicción de valores futuros.

1.2.1. Componentes de una serie de tiempo: Las series de tiempo tienen cuatro componentes: de tendencia, cíclico, estacional e irregular.

- De tendencia³³: Aunque los datos de las series de tiempo suelen mostrar fluctuaciones aleatorias, las series de tiempo también muestran un desplazamiento o movimiento gradual hacia valores relativamente altos o bajos a través de un lapso largo. A este desplazamiento gradual de la serie de tiempo se le conoce como la tendencia de la serie de tiempo
- Componente cíclico: Son fluctuaciones que se observan en la serie de tiempo, puntos que se van hacia arriba y hacia debajo de la tendencia de forma regular, “Toda sucesión recurrente de puntos que caiga abajo y arriba

³² ANDERSON, D.R.; SWEENEY, D. J. y WILLIAMS T. A. Estadística para Administración y Economía. Mexico: Caengage Learning 10a. edición, 2008. p. 767

³³ Ibid p. 767

de la línea de tendencia y que dure más de un año puede atribuirse al componente cíclico de la serie de tiempo”³⁴.

- **Componente estacional:** Son patrones que se observan en periodos de tiempo largos, determinados claramente por factores externos que se repiten con cierta frecuencia, como por ejemplo la el aumento en la venta de paraguas en las épocas lluviosas del año.
- **Componente irregular:** Es la componente impredecible de la serie de tiempo, se le atribuye a factores que son imprevistos y no recurrentes, se comporta como un ruido blanco, es decir aleatorio y con media cero.

1.2.2. Métodos de pronóstico: Según Gujarati, D.³⁵ hay 5 enfoques generales para la realización de pronósticos de producción. A continuación, se presenta una breve explicación de cada una, y más adelante se explicará la metodología ARIMA, de interés principal para el presente trabajo de grado.

1.2.2.1. Métodos de suavizamiento exponencial:

$$F_{i+1} = \alpha * Y_i + (1 - \alpha) * F_i \quad (\text{Ec. 1.2.1})$$

Es un promedio ponderado de los últimos n datos de la serie de tiempo donde “sólo hay que elegir un peso, el peso para la observación más reciente. Los pesos para

³⁴ Ibid p. 769

³⁵ GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. 2009. Cap. 22, pág 774.

los demás datos se calculan automáticamente y son más pequeños a medida que los datos son más antiguos.”³⁶.

1.2.2.2. Modelos de regresión uniecuacionales: Es un análisis de regresión que tiene como resultado una sola ecuación que describe la componente tendencial. Puede ser con respecto a una sola variable (Ejemplo las curvas empíricas de Arps que dependen solo del tiempo como variable independiente), o multivariable.

1.2.2.3. Modelos de regresión de ecuaciones simultáneas³⁷: Se utilizan para sistemas donde hay variables que son mutuamente dependientes (endógenas) y otras que no (exógenas), es decir cada una depende de la otra, entonces debe haber una ecuación para cada variable endógena.

1.2.2.4. Modelos autorregresivos integrados de promedios móviles (ARIMA)³⁸: “La publicación de G. P. E. Box y G. M. Jenkins ‘Time Series Analysis: Forecasting and Control’, marcó el comienzo de una nueva generación de herramientas de pronóstico. Popularmente conocida como metodología de Box-Jenkins (BJ), pero técnicamente conocida como metodología ARIMA, el interés de estos métodos de pronósticos no está en la construcción de modelos uniecuacionales o de ecuaciones simultáneas, sino en el análisis de las propiedades

³⁶ ANDERSON, D.R.; SWEENEY, D. J. y WILLIAMS T. A. Estadística para Administración y Economía. Mexico: Caengage Learning 10a. edición, 2008. p. 774

³⁷ GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. Cap. 22, Capítulo 18.

³⁸G.P.E. Box y G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1978.

probabilísticas, o estocásticas, de las series de tiempo económicas por sí mismas según la filosofía de ‘que los datos hablen por sí mismos’.”³⁹

1.2.2.5. Modelos de vectores autorregresivos (VAR)⁴⁰: Es similar al modelo de ecuaciones simultaneas, salvo que, en este, las variables dependen de los valores pasados de sí mismas y de las otras.

1.3. SERIES ESTACIONARIAS EN DIFERENCIAS Y EN TENDENCIAS

Antes de hablar de la metodología ARIMA, es conveniente mencionar que existen series estacionarias en tendencias y en diferencias. Un modelo ARIMA(p,d,q) supone que la serie que se maneja es estacionaria en las primeras “d” diferencias, pero puede que sea estacionaria en tendencia. De ser así, se debe retirar la componente tendencial de la serie para luego aplicar la metodología de Box & Jenkins.

Los procesos estacionarios en diferencias y en tendencia se pueden identificar de la siguiente manera⁴¹:

- Procesos estocásticos estacionarios en diferencias (PED): Un proceso es estacionario en diferencias si presenta tendencia estocástica, esto es, variable e impredecible.

$$y_t = y_{t-1} + u_t$$

Ec. 1.3.1.

³⁹ GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. Cap. 22, pág 774.

⁴⁰ Ibíd., pág. 784 – 797.

⁴¹ Ibíd., pág 745.

En estos procesos la componente tendencial se elimina después de d diferencias.

$$\Delta y_t = y_t - y_{t-1} = u_t \quad \text{Ec. 1.3.2.}$$

“A u_t se le conoce como serie de tiempo sin tendencia”⁴²

- Procesos estacionarios en tendencia (PET): Estos procesos presentan tendencia determinista, es decir, la componente tendencial es totalmente predecible y no variable.

$$y_t = B_1 + B_2 * t + u_t \quad \text{Ec. 1.3.3.}$$

Donde u_t es el residuo o la diferencia entre el valor medido y la media.

De esta manera es posible pronosticar la media sin problema, ya que puede describirse en función del tiempo ($B_1 + B_2 * t$).

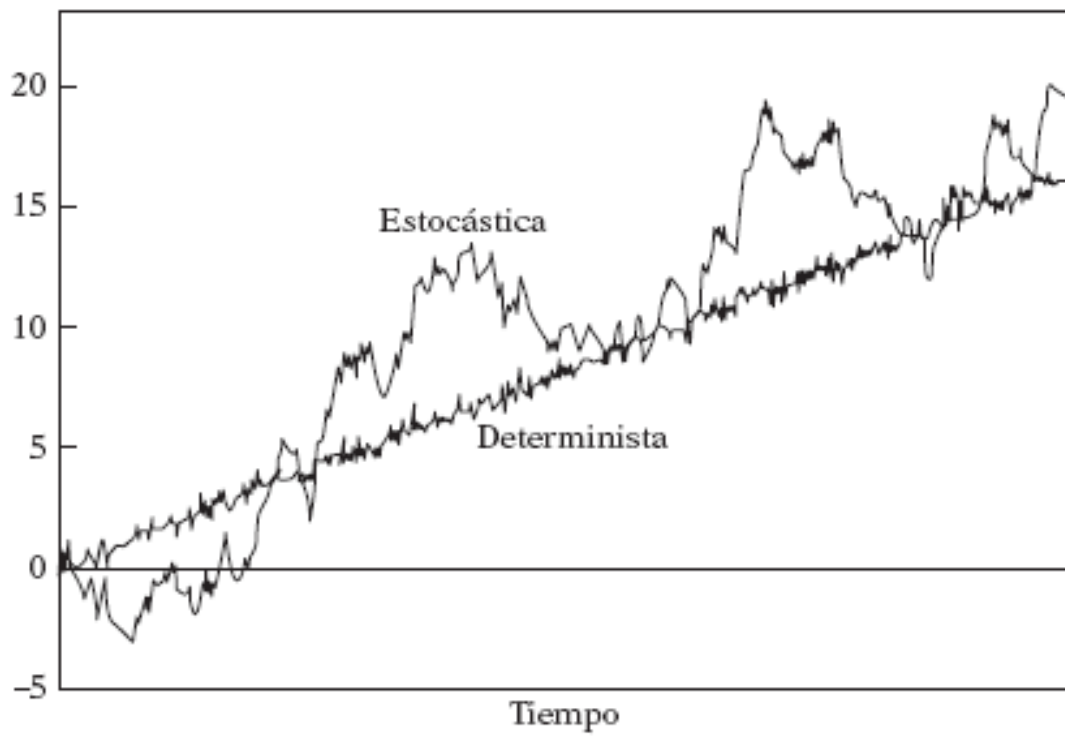
Al despejar los residuos, se obtiene una serie estocástica y estacionaria:

$$u_t = y_t - B_1 - B_2 * t \quad \text{Ec. 1.3.3.}$$

Cabe mencionar que esta transformación aplica también para distintos tipos de tendencia (cuadrática, cubica, exponencial, etc.)

⁴² Ibíd. Cap 21, pág 761

Figura 1.3.1: Tendencia estocástica y determinista.⁴³



⁴³ Charemza et al., *New Directions in Econometric Practice: A General to Specific Modelling, Cointegration and Vector Autoregression*, p. 91. Citado por: GUJARATI, Damodar. PORTER, Dawn.

Fuente: Charemza et al., New Directions in Econometric Practice: A General to Specific Modelling, Cointegration and Vector Autoregression, p. 91. Citado por: GUJARATI, Damodar. PORTER, Dawn.

1.4. METODOLOGIA DE BOX & JENKINS (ARIMA)⁴⁴

El modelo Autorregresivo Integrado de Media Movil, ARIMA de orden (p,d,q) consta de tres partes fundamentales:

- La parte autorregresiva de orden p.
- La parte de media móvil de orden q.
- El grado de diferenciación de la serie, d.

Se debe aclarar que un modelo AR(p) es un modelo ARIMA(p,0,0), de igual manera un modelo MA(q) es equivalente a ARIMA(0,0,q), y un modelo ARMA(p,q) a ARIMA(p,0,q).

El modelo ARIMA(p,d,q) se representa matemáticamente de la siguiente manera:

$$z_t = \delta + \phi_1 z_{t-1} + \dots + \phi_p z_{t-p} - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} + a_t \quad (\text{Ec. 1.3.7})$$

Donde:

- z es la serie de las diferencias de orden d de la serie de tiempo original.
- δ es un término constante equivalente a $\delta = \bar{z} * (1 - \phi_1 - \dots - \phi_p)$. Si la media es 0, $\delta = 0$

⁴⁴ G.P.E. Box y G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1978.

- $\phi_1, \phi_2 \dots \phi_p$ son los coeficientes de la parte autorregresiva del modelo.
- $\theta_1, \theta_2 \dots \theta_p$ son los coeficientes de la parte de media móvil del modelo.
- a corresponde a los residuos obtenidos para cada tiempo.

Y la predicción puntual es:

$$\hat{z}_t = \delta + \phi_1 z_{t-1} + \dots + \phi_p z_{t-p} - \hat{\theta}_1 \hat{a}_{t-1} - \dots - \hat{\theta}_q \hat{a}_{t-q} + \hat{a}_t \quad (\text{Ec. 1.3.8})$$

Donde:

- $\hat{a}_{t-1} = a_{t-1} = z_{t-1} - \hat{z}_{t-1}$ en el caso de que exista la medición z_{t-1} que permita determinarlo, en caso contrario $\hat{a}_{t-1} = 0$.
- $\hat{a}_t = 0$ en la predicción, debido a la inexistencia de la medición z_t .
- $a_t = z_t - \hat{z}_t$

Según lo resumen BOWERMAN, Bruce, et al.⁴⁵ La metodología de Box-Jenkins es un procedimiento iterativo de 4 pasos:

- Identificación tentativa por medio de las funciones de auto correlación simple y parcial (FAS y FAP respectivamente).
- Estimación de los parámetros del modelo.
- Comprobación y diagnóstico, que en el presente libro será realizado por el método del valor P.
- Pronóstico.

⁴⁵ BOWERMAN, Bruce. O'CONNELL, Richard. KOEHLER, Anne. Pronósticos, series de tiempo y regresión, un enfoque aplicado. CENGAGE Learning. Cuarta edición. 2006. Capítulo 9. Página 401

1.4.1. Identificación tentativa: En este paso se pretende identificar el orden del modelo. La identificación tentativa del modelo ARIMA se realiza graficando la función de autocorrelación simple (FAS, que para efectos de cálculo será llamada r_k) y la función de autocorrelación parcial (FAP o r_{kk}).

1.4.1.1. Función de autocorrelación simple: “Esta cantidad mide la relación lineal entre las observaciones de la serie temporal separadas por un desfaseamiento de k unidades de tiempo”⁴⁶. El valor de r_k siempre está entre -1 y 1.

La FAS en el desfaseamiento k es:

$$r_k = \frac{\sum_{t=b}^{n-k} (z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=b}^n (z_t - \bar{z})^2} \quad (\text{Ec. 1.3.1})$$

La desviación estándar de r_k es:

$$s_{r_k} = \frac{1}{(n-b+1)^{1/2}} \quad \text{Para } k = 1 \quad (\text{Ec. 1.3.2})$$

$$s_{r_k} = \frac{(1 + 2 \sum_{j=1}^{k-1} r_j^2)^{1/2}}{(n-b+1)^{1/2}} \quad \text{Para } k > 1 \quad (\text{Ec. 1.3.3})$$

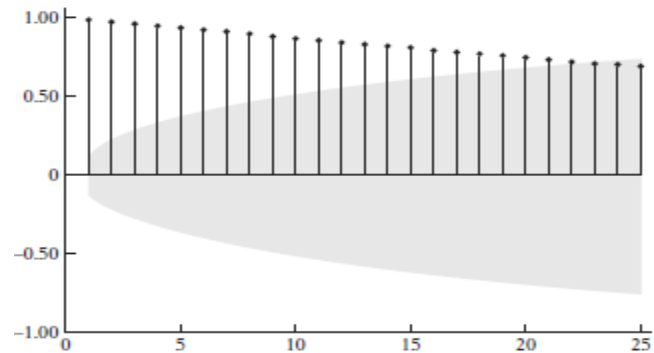
Donde:

- \bar{z} es la media aritmética de todos los datos de la serie de tiempo
- b corresponde a la posición del primer dato de la serie de tiempo (1 cuando no ha sido alterada, 2 cuando son primeras diferencias, etc.)

Con esto se realiza la gráfica de FAS vs. k que tiene la siguiente forma:

⁴⁶ Ibíd. Capítulo 9. Página 407

Figura 1.3.2: Función de autocorrelación simple.



Fuente: GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. Cap. 22, pág 779.

La franja gris creciente que se observa, son los valores de $2 * s_{r_k}$, y es una herramienta que ayuda a definir el orden del modelo.

1.4.1.2. Función de autocorrelación parcial: Este valor es la función de autocorrelación que tienen las dos observaciones separadas por el desfase k , sin tener en cuenta los efectos de las observaciones que hay entre ellas.

$$r_{kk} = r_1$$

Para $k = 1$ (Ec. 1.3.4)

$$r_{kk} = \frac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} * r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} * r_j} \quad \text{Para } k > 1 \quad (\text{Ec. 1.3.5})$$

$$s_{r_{kk}} = \frac{1}{(n-b+1)^{1/2}} \quad (\text{Ec. 1.3.6})$$

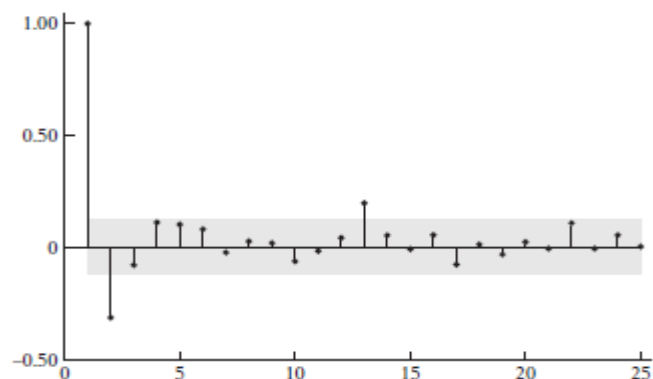
Donde:

- $r_{kj} = r_{k-1,j} - r_{kk} * r_{k-1,k-j}$, Para $j = 1, 2, \dots, k-1$.
- b Corresponde a la posición del primer dato de la serie de tiempo (1 cuando no ha sido alterada, 2 cuando son primeras diferencias, etc.)
- $s_{r_{kk}}$ es la desviación estándar.

Con esto se realiza la gráfica de FAP vs. k que tiene la forma que se presenta en la Figura 1.3.3.

La franja gris que se observa, son los valores de $2 * s_{r_{kk}}$, y es una herramienta que ayuda a definir el orden del modelo.

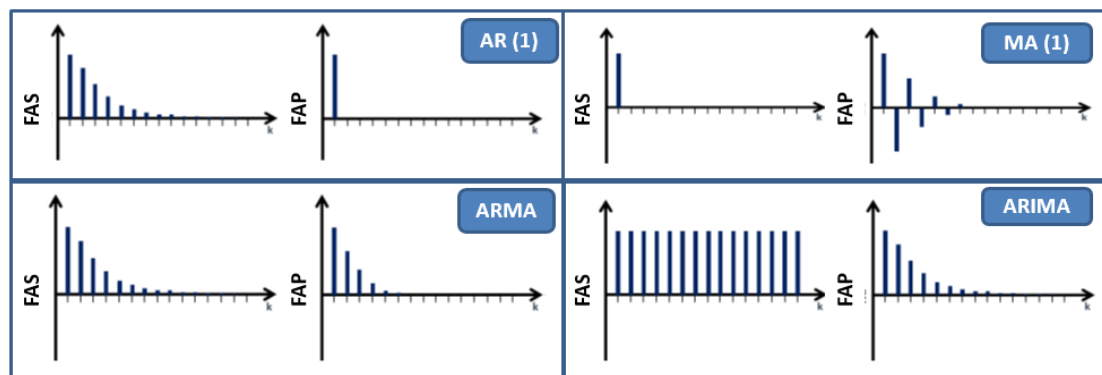
Figura 1.3.3: Función de autocorrelación parcial.



Fuente: GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. Cap. 22, pág 779.

A continuación se presentan algunas gráficas de FAS y FAP comunes para determinado tipo de modelo:

Figura 1.3.4: FAS y FAP típicas para determinados modelos.



Fuente: NOVALES, Alfonso. Econometría, 2 ed. Madrid: McGraw Hill, 1993.

En los casos de los modelos ARMA y ARIMA, la identificación del orden del modelo no es tan sencilla, por tanto se recurre a la determinación por prueba y error, basándose en el análisis de los residuos para encontrar el mejor modelo, como se explicará más adelante.

1.4.2. Estimación de los coeficientes del modelo⁴⁷: La estimación de los coeficientes del modelo se realiza mediante una regresión no lineal bastante compleja, pero es realizada por distintos software, entre los que se destaca el lenguaje de programación estadístico R, por ser muy robusto y gratuito.

La función que se emplea en R es `arima()`⁴⁸, que recibe como parámetros el vector de datos, y el vector de orden (`order = c(p,d,q)`).

Por ejemplo, para encontrar un modelo ARIMA(2,1,3), que describa el vector de datos “Datos”, la línea en R sería:

- `arima(Datos, order = c(2,1,3))`

1.4.3. Comprobación y diagnóstico: Para la comprobación del modelo se realiza un análisis de los residuos, empleando el *valor probabilístico (Valor P)*, que se obtiene de un análisis de los residuos \hat{a} obtenidos del modelo.

Es necesario utilizar “una estadística o dato numérico que determine si las primeras k autocorrelaciones muestrales de los residuos, considerados juntos, indican suficiencia del modelo”⁴⁹. El dato numérico a emplear es el de Ljung-Box:

$$Q^* = n'(n' + 2) \sum_{l=1}^k (n' - 1)^{-1} r_l^2(\hat{a})$$

Donde:

- $n' = n - d$, siendo n el número de datos, y d el grado de diferenciación de la serie de tiempo, siendo equivalente decir que $n' = n - b + 1$.

⁴⁷ R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

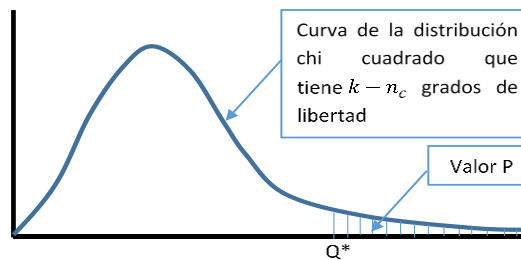
⁴⁸ B. D., Ripley (2002). Time series in R 1.5.0. *R News*, **2/2**, 2–7. Disponible en: https://www.r-project.org/doc/Rnews/Rnews_2002-2.pdf.

⁴⁹ BOWERMAN, Bruce. O'CONNELL, Richard. KOEHLER, Anne. Pronósticos, series de tiempo y regresión, un enfoque aplicado. CENGAGE Learning. Cuarta edición. 2006. Capítulo 10. Página 458.

- $r_t^2(\hat{a})$ es el cuadrado de la Función de autocorrelación simple (FAS) de los residuos \hat{a}
- k es un valor arbitrario, normalmente seis o un múltiplo de seis, según se necesite.

Después de calcular este valor, se procede a calcular el *Valor P*, que es “el área bajo la curva de la distribución chi cuadrado que tiene $k - n_c$ grados de libertad, a la derecha de Q^* ”.⁵⁰

Figura 1.3.5: Valor P.



Fuente: BOWERMAN, Bruce. O’CONNELL, Richard. KOEHLER, Anne. Pronósticos, series de tiempo y regresión, un enfoque aplicado. CENGAGE Learning. Cuarta edición. 2006. Capítulo 10. Página 460.

El análisis mencionado también es desarrollado por el lenguaje de programación R, mediante la función `Box.test()` que recibe como parámetros el vector de residuos del modelo, y el dato estadístico a usar que puede ser elegido entre el de Ljung-Box, o el de Box-Pierce (`type = "Ljung-Box"`, o, `type = "Box-Pierce"`).⁵¹

⁵⁰ *Ibíd.* Capítulo 10. Página 459.

⁵¹ TRAPLETTI, A., for: R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Como otro parámetro de evaluación se tiene el Criterio de Información de Akaike, que obtiene un valor que indica la pérdida de información de modelo y castiga al modelo por el número de parámetros, permitiendo así, mediante iteración y comparación encontrar un modelo que describa los datos con el número de parámetros indicado.⁵²

1.4.4. Pronóstico: El pronóstico se realiza punto por punto con la ecuación 1.3.8, teniendo en cuenta que los valores futuros de los residuos siempre van a ser 0, debido a que no hay datos reales para comparar.

Este proceso es realizado por el software R, mediante la función `predict()`⁵³, proporcionando como parámetros el modelo y el número de valores futuros a pronosticar.

Como alternativa, está la función `forecast()`⁵⁴ del paquete `forecast`, que predice además intervalos de confianza de 95% y 75%.

2. HERRAMIENTAS EMPLEADAS PARA EL DESARROLLO Y FUNCIONAMIENTO INTERNO DEL SOFTWARE

⁵² R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

⁵³ *Ibíd.*

⁵⁴ HYNDMAN, R. J., for: R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Para el desarrollo del software presentado se utilizaron cuatro herramientas principales:

- Visual Studio Community 2015
- R
- R.Net
- Paquete de interoperabilidad entre Excel y Visual Studio

2.1. VISUAL STUDIO COMMUNITY 2015:

Visual Studio en su versión “Community” (ver Figura 2.1.1) es un entorno de desarrollo integrado ofrecido gratuitamente por Microsoft para pequeños programadores (grupos de menos de cinco personas), funciona con la plataforma .Net que cuenta con distintos lenguajes de programación disponibles entre los cuales se encuentra C#, que fue seleccionado para el desarrollo del presente trabajo.

Entre las principales bondades de este IDE (Integrated Development Enviroment), se encuentran:

- El paquete System.Windows.Forms: Es útil para la creación de los formularios y las partes centrales de las aplicaciones
- Windows Presentation Fundation (WPF): Permite desarrollar ventanas personalizadas con buen nivel de gráficos utilizando Blend for Visual Studio.
- Herramientas de depuración.

- Administrador de paquetes NuGet: Permite instalar paquetes gratuitos creados por la comunidad o por los mismos desarrolladores de Microsoft. Para los objetivos del presente trabajo de grado los paquetes más importantes son R.Net (que permite ejecutar un motor de R en segundo plano) y el paquete de interoperabilidad con Excel, útil para generar informes automáticamente.

2.2. R⁵⁵

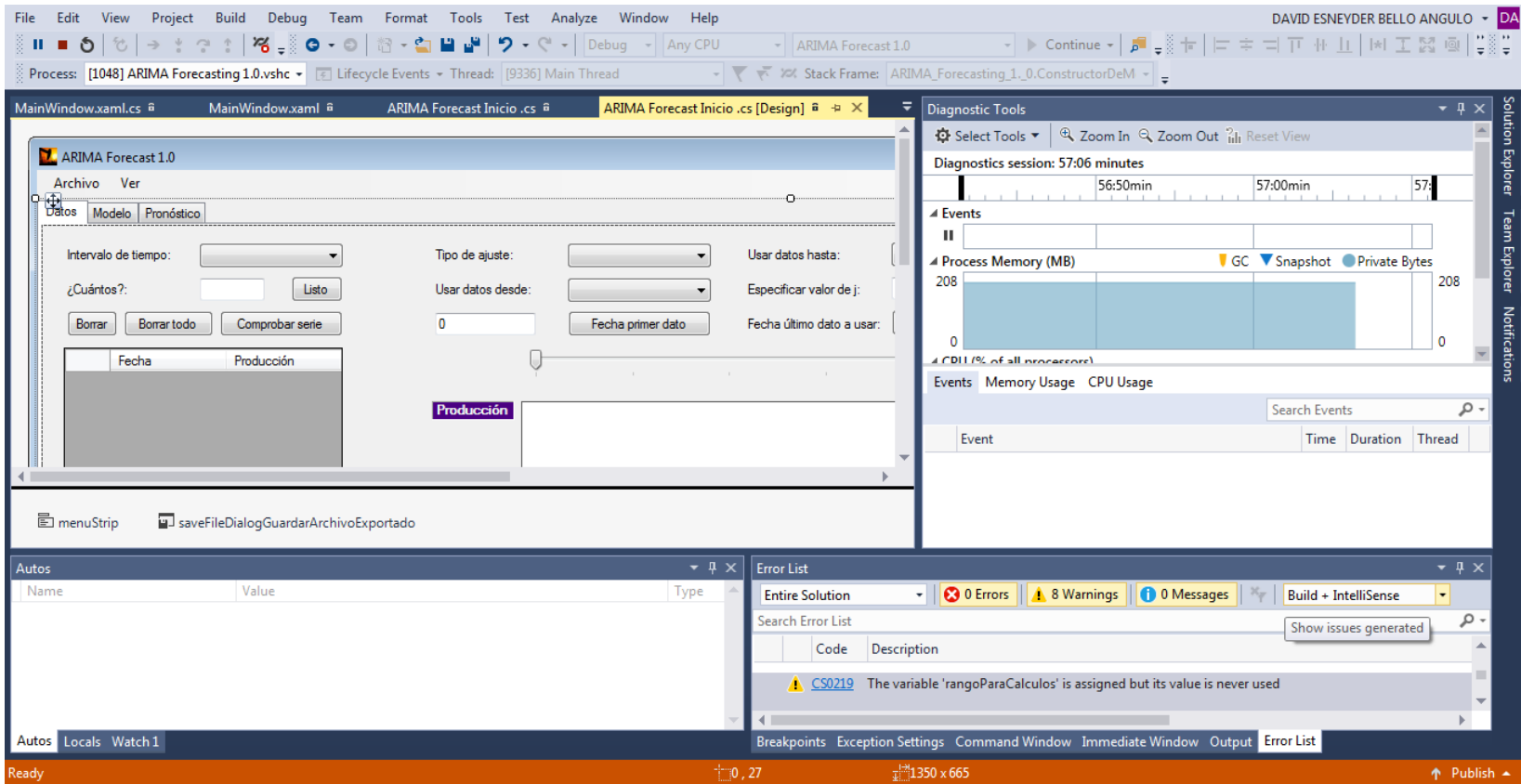
R es un software y lenguaje de programación ampliamente utilizado en el área estadística por su gran cantidad de librerías y funciones que facilitan el análisis de datos, además de proporcionar gráficos de alta calidad.

R es un software libre protegido por licencia GNU – General Public License, que garantiza que sea gratuito y de código abierto.

Una de sus grandes fortalezas es que se permite la interacción con diversos programas y lenguajes de programación.

⁵⁵ R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Figura 2.1.1: IDE Visual Studio Community.



Fuente: Visual Studio Community 2015. Entorno de desarrollo integrado. Microsoft. Redmond, Washington, Estados Unidos. URL <https://www.visualstudio.com/>.

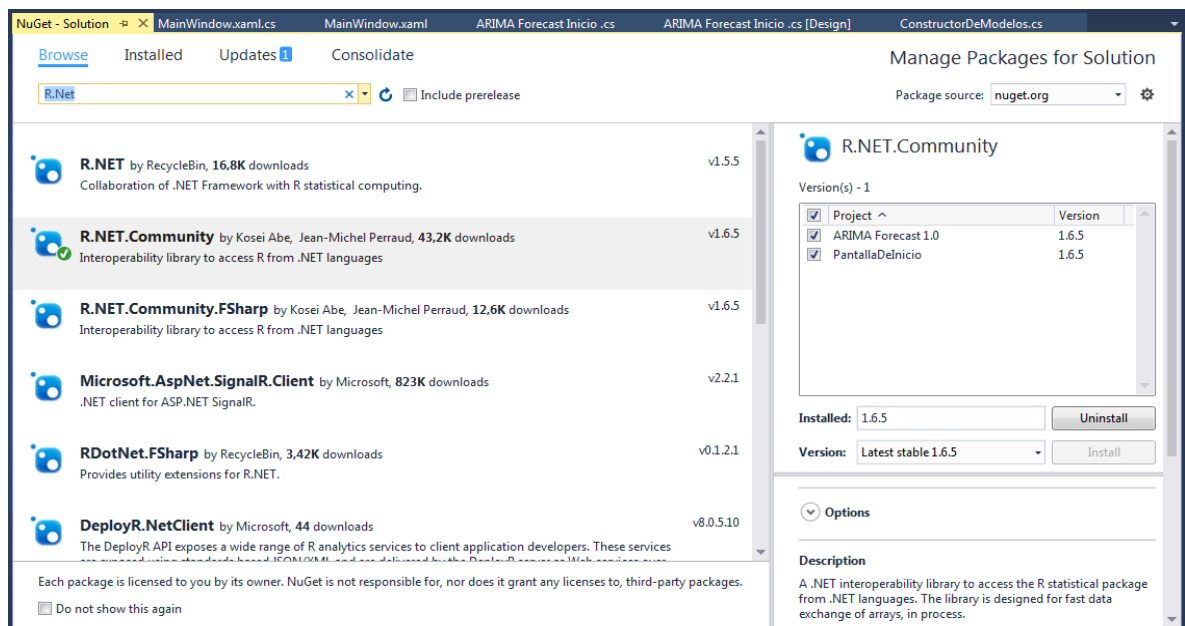
2.3. R.NET

El paquete de interoperabilidad entre R y la plataforma .Net, R.Net, es software libre disponible para su descarga en NuGet.org. Para abrir NuGet Package Manager desde Visual Studio se debe seguir la ruta Tools/NuGet Package Manager /Manage NuGet Packages for Solution. Una vez abierto, se selecciona “Browse” y se busca el paquete R.Net.Community (ver Figura 2.2.1).

El paquete es tomado como una clase, luego de instalada debe ser llamada (`using RDotNet;`) y luego debe crearse una instancia de clase de la siguiente forma para poder utilizarlo:

- `REngine motorDeR = REngine.GetInstance();`

Figura 2.2.1: Descarga de R.Net desde NuGet.



Fuente: Visual Studio Community 2015. Entorno de desarrollo integrado. Microsoft. Redmond, Washington, Estados Unidos. URL <https://www.visualstudio.com/>.

Esta clase tiene sub clases que permiten relacionar los objetos de R con los objetos de .Net, tales como vectores y matrices. La tabla 2.2.1 relaciona los objetos de .Net con los de R.Net.

Es necesario tener en cuenta que tanto para los enteros como para decimales, el mínimo valor en R es $-2^{31} + 1$ mientras que en .NET es -2^{31} .

También es importante saber que mientras en .Net y R.Net los índices de vectores y matrices parten de la posición cero y van hasta la longitud menos uno, en R parten de uno y van hasta la longitud del arreglo.

Tabla 2.1.1: Objetos de R, R.Net y .Net relacionados según su tipo.

R	R.NET	.NET Framework
character vector	RDotNet.CharacterVector	System.String[]
integer vector	RDotNet.IntegerVector	System.Int32[]
real vector	RDotNet.NumericVector	System.Double[]
complex vector	RDotNet.ComplexVector	System.Numerics.Complex[]
raw vector	RDotNet.RawVector	System.Byte[]
logical vector	RDotNet.LogicalVector	System.Boolean[]
character matrix	RDotNet.CharacterMatrix	System.String[,]
integer matrix	RDotNet.IntegerMatrix	System.Int32[,]
real matrix	RDotNet.NumericMatrix	System.Double[,]
complex matrix	RDotNet.ComplexMatrix	System.Numerics.Complex[,]
raw matrix	RDotNet.RawMatrix	System.Byte[,]
logical matrix	RDotNet.LogicalMatrix	System.Boolean[,]
list	RDotNet.GenericVector	
data frame	RDotNet.GenericVector	
data frame	RDotNet.DataFrame	
function	RDotNet.Function	
factor	RDotNet.Factor	System.Int32[]

2.3.1. Funciones principales: Una vez creada la instancia de clase (motorDeR en este caso), puede accederse a las funciones que trae programadas:

2.3.1.1. SetSymbol(): Es la función que se utiliza para pasar objetos (vectores, enteros, cadenas de caracteres, entre otros) desde .Net a R. Cabe aclarar que solo se permiten objetos de R.Net (como NumericVector), por lo tanto si por ejemplo, se quiere pasar un objeto del tipo double[], primero debe pasarse todos los ítems del double[] al NumericVector, y pasar este último a R. Como parámetros recibe una cadena de caracteres que es el nombre que se le dará al objeto en R y el objeto de R.Net. Ejemplo:

- `motorDeR.SetSymbol("Datos", datosParaR);`

2.3.1.2. Evaluate(string): Permite ingresar cadenas de caracteres a R que indiquen acciones que serán ejecutadas como si se escribiera directamente en la consola, por ejemplo:

- `motorDeR.Evaluate("modeloArima <- auto.arima(Datos)");`

La línea anterior realiza un modelo buscando entre distintos modelos el mejor AIC, con los datos previamente ingresados al motor de R y lo guarda con el nombre modeloArima, para que luego pueda ser utilizado.

Otra utilidad de esta función, es pasar datos de R a .Net, ejemplo:

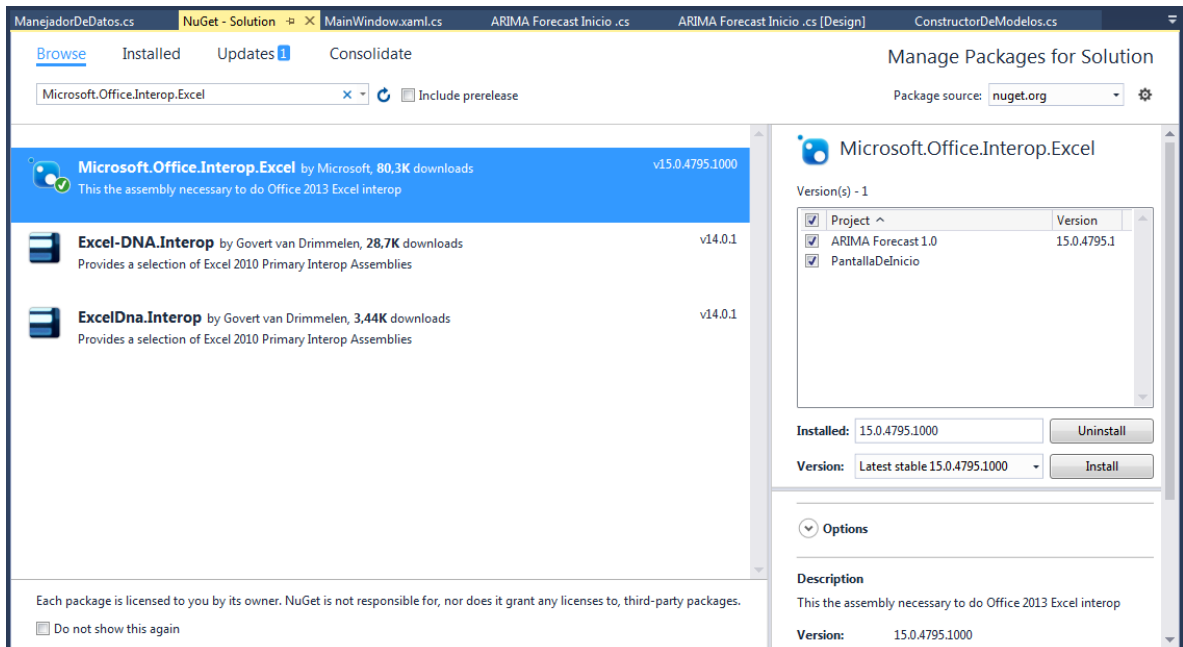
- `double aic =`

```
Convert.ToDouble(motorDeR.Evaluate("modeloArima$aic").AsNumeric().First());
```

2.4. PAQUETE DE INTEROPERABILIDAD ENTRE MICROSOFT EXCEL Y VISUAL STUDIO

También está disponible en NuGet.org como “Microsoft.Office.Interop.Excel” (ver Figura 2.4.1).

Figura 2.4.1: Paquete de interoperabilidad entre Excel y Visual Studio.



Permite crear y modificar archivos de Excel, y debe ser llamado también para poder ser utilizado:

- `using Excel = Microsoft.Office.Interop.Excel;`

Se utiliza "Excel =" para abreviar el nombre ya que tendrá que ser utilizado más Adelante.

Para crear un archivo de Excel es necesario crear una instancia de la clase, el libro y la hoja que van a ser utilizados, de la siguiente manera:

- `Excel.Application excel = new Excel.Application();`
- `Excel._Workbook libro = null;`
- `Excel._Worksheet hoja = null;`
- `libro = (Excel._Workbook)excel.Workbooks.Add(Excel.XIWBATemplate.xIWBATWorksheet);`
- `hoja = (Excel._Worksheet)libro.Worksheets.Add();`

Se debe también crear un objeto de la sub clase Rango, que es el que permite especificar con que celda o celdas se va a trabajar en determinado momento.

- `Excel.Range rango = null;`

El rango puede ser seleccionado de distintas maneras, por ejemplo:

- `rango = excel.Columns[3];`
- `rango = hoja.Range["B4", "C4"];`

El contenido de una celda se cambia de la siguiente manera:

- `hoja.Cells[4, 2] = "Fecha";`
- `hoja.Cells[6, 5] = 581,2;`

También es posible crear gráficos de manera automática, por ejemplo, las líneas típicas para crear un gráfico de líneas son:

- Excel.ChartObjects coleccionDeGraficos = null;
- Excel.ChartObject contenedorDeGrafico = null;
- Excel.Chart grafico = null;
- Excel.Range rangoGrafico = null;
- Excel.Range rangoParaCalculos = null;
- coleccionDeGraficos =
(Excel.ChartObjects)hoja.ChartObjects(Type.Missing);
- contenedorDeGrafico = (Excel.ChartObject)coleccionDeGraficos.Add(415,
61.5, 400, 250); //posición (izquierda, arriba), ancho y alto del cuadro del
gráfico
- grafico = contenedorDeGrafico.Chart;
- rangoGrafico = hoja.Range["B4", "C20"];
- grafico.ChartType = Excel.XlChartType.xlLine;
- grafico.HasTitle = true;
- grafico.ChartTitle.Caption = "Pronóstico de producción";
- grafico.ApplyDataLabels(Excel.XlDataLabelsType.xlDataLabelsShowValue);
- grafico.SetSourceData(rangoGrafico);

Una vez creado el archivo de Excel se debe guardar de la siguiente manera:

- libro.SaveAs(@"D:\Pronóstico de producción.xlsx");

El parámetro debe ser una cadena de caracteres con la ruta en que se desea que el archivo sea guardado.

2.5. FUNCIONAMIENTO INTERNO DEL SOFTWARE

La guía para realizar los procedimientos planteados se encuentra en el manual de usuario del software.

2.5.1. Carga de datos:

- Carga de datos desde un archivo o copiados desde Excel.
- Se evalúa la continuidad de la serie de tiempo dependiendo del intervalo que el usuario elija y se rellenan los datos faltantes con cero.
- Se ha planteado el método de media móvil para el ajuste de la serie, que trabaja únicamente con los datos cero, dándoles el valor del promedio de los siete datos anteriores.
- Se permite al usuario elegir el rango de datos con el que desea trabajar, con el fin de que ignore los datos no representativos (antes del inicio de la declinación, por ejemplo) y que elija los que mejor representen la tendencia actual de declinación.

2.5.2. Modelos usados: Como alternativas para la construcción de modelos se emplearon dos categorías, las curvas de declinación de Arps (Exponencial, Hiperbólica y Armónica) y la metodología ARIMA (de Box & Jenkins) aplicada a la elaboración de pronósticos en el por Montes Erik en “Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos”⁵⁶ que fue tomado como base del presente trabajo.

2.5.2.1. Curva exponencial y curva armónica de Arps: La metodología planteada para la solución de las curvas exponencial y armónica es la misma. Hay tres opciones:

- Especificar el valor de Di: La producción inicial “qi” toma el valor del primer dato seleccionado, y el valor de la declinación inicial “Di” es ingresado por el usuario.
- Hacer que la curva pase por los puntos inicial y final: La producción inicial “qi” toma el valor del primer dato seleccionado, y para hallar la declinación inicial se reemplaza el último dato (producción final “qf” y tiempo final “tf”) y se despeja, quedando:

Para curva exponencial:
$$Di = \frac{\ln(\frac{qi}{qf})}{tf} \quad \text{Ec. 2.5.1.}$$

Para curva Armónica:
$$Di = \frac{\frac{qi}{qf} - 1}{tf} \quad \text{Ec. 2.5.2.}$$

⁵⁶ MONTES, Erick. Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos. Trabajo de grado (Magíster en Ingeniería de Hidrocarburos) Universidad Industrial de Santander. Bucaramanga. 2016.

- Solucionar las ecuaciones con mínimos cuadrados ordinarios, volviéndolas lineales de la forma $y = a + b * x$:

Para curva exponencial: $\ln(q) = \ln(q_i) - Di * t$ Ec. 2.5.3.

Para curva Armónica: $\frac{1}{q} = \frac{1}{q_i} + \frac{Di}{q_i} * t$ Ec. 2.5.4.

Sea la matriz “m” una matriz con dos columnas donde la primera está llena de unos (ya que el coeficiente “a” o intercepto siempre está multiplicado por uno) y la segunda corresponde a la variable independiente (que multiplica al coeficiente “b” o pendiente) (ver Tabla 2.5.1), y sea “y” el vector de la variable dependiente (ver Tabla 2.5.2).

Tabla 2.5.1: Matriz “m” para la regresión por mínimos cuadrados.

	Tiempo
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12
1	13
1	14

1	15
1	16

La solución por mínimos cuadrados indica que el vector “c” de coeficientes (donde c_1 es el intercepto “a”, y c_2 la pendiente “b”) se determina con la siguiente ecuación:

$$c = (T(m) * m)^{-1} * (T(m) * y) \quad \text{Ec. 2.5.5.}$$

Donde $T(m)$ es la traspuesta de la matriz y $(T(m) * m)^{-1}$ es la inversa de la matriz resultante de multiplicar la matriz “m” por su traspuesta.

Tabla 2.5.2: Vector “y” para la regresión por mínimos cuadrados

y
8
7
9
6
7
4
8
6
5
7
4
5
6
3
4
5

La ecuación anterior es solucionada con la ayuda de R.Net (habiendo ingresado de manera previa la matriz “m” y el vector “y” con la función SetSymbol()) con la línea:

- `motorDeR.Evaluate("y <- solve(t(m) %% m) %% (t(m) %% y)");`

2.5.2.2. Curva hiperbólica de Arps: Se plantean tres opciones principales:

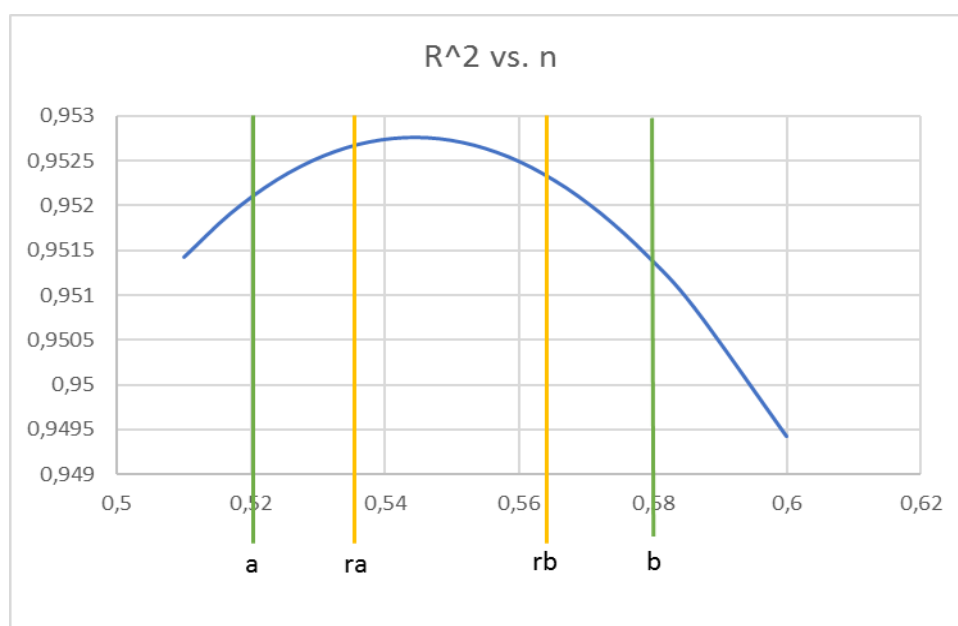
- Manual: La producción inicial “qi” toma el valor del primer dato seleccionado, y el valor de la declinación inicial “Di” y del exponente “n” es ingresado por el usuario.
- Semiautomático: Se obliga al modelo a pasar por el punto final de los datos (producción final “qf” a tiempo “tf”). La producción inicial “qi” toma el valor del primer dato seleccionado, el valor del exponente “n” es ingresado por el usuario y con base en esto se despeja la declinación inicial “Di” para calcularla:

$$Di = \frac{\left(\frac{qi}{qf}\right)^{n-1}}{n*tf} \quad \text{Ec. 2.5.6.}$$

- Automático: Se obliga al modelo a pasar por el punto final de los datos (producción final “qf” a tiempo “tf”). La producción inicial “qi” toma el valor del primer dato seleccionado, el valor del exponente “n” se va variando (calculando cada vez la declinación inicial “Di” respectiva) con un algoritmo de iteraciones (ver Figura 2.5.1) diseñado para encontrar el máximo valor del coeficiente de determinación R^2 .

En inicio se itera desde un “n” mínimo dado por el usuario, en caso de que no se seleccione es cero (cuando n es cero, el modelo hiperbólico se trata como uno exponencial) a un paso determinado hasta que R^2 comience a disminuir, acto seguido se definen los valores de los límites “a” y “b” alrededor de donde está el punto máximo, y se comienza a “escalar” la curva por los dos lados, equilibrando, para procurar que el punto máximo esté siempre en medio.

Figura 2.5.1: Explicación del algoritmo para encontrar el mejor modelo



Se establece que:

$$r_a = a + \frac{1}{4} * (b - a) \quad \text{Ec. 2.5.7.}$$

$$r_b = a + \frac{3}{4} * (b - a) \quad \text{Ec. 2.5.8.}$$

Después se procede a comparar r_a y r_b para asignar nuevos valores a los límites, de la siguiente manera:

- Si $r_a > r_b$ entonces $b = r_b$
- Si $r_a < r_b$ entonces $a = r_a$

Al final de las iteraciones se calcula el valor de n :

$$n = \frac{a+b}{2} \quad \text{Ec. 2.5.9.}$$

2.5.2.3. Metodología ARIMA de Box & Jenkins: La metodología ARIMA es realizada en R con la ayuda de R.Net. Se proporcionan dos métodos principales:

- Usar función `auto.arima()`: La función `auto.arima` de la librería `forecast`, si no se le pasan más parámetros que la serie de tiempo, realiza una iteración para encontrar un buen modelo con base en el criterio de Información de Akaike y el criterio de Información de Bayes.
- Definir el orden del modelo ARIMA: se usa la función `arima()` que acepta como parámetros la serie de tiempo y un vector de orden (`order = c(p,d,q)`), o la función `auto.arima()`, pasando como parámetros restricciones (`max.p, max.q, start.p, start.q, d`).
por ejemplo, para realizar un modelo ARIMA(5,1,4) (con la ayuda de R.Net, utilizando datos que ya estén cargados en R, y habiendo especificado con anterioridad el valor de p , d y q) y guardarlo en la variable `modelo` se escribe lo siguiente:

```
motorDeR.Evaluate("modeloArima <- auto.arima(" +
nombreDelVectorDeDatosR + ", max.p = " + p + ", max.q = " + q + ", start.p =
" + p + ", start.q = " + q + ", d=" + d + ")");
```

- Mejor modelo ARIMA por AIC: Se calcula el criterio de información de Akaike para todos los modelos ARIMA(p,d,q) existentes entre el ARIMA (1,1,1) y el ARIMA (9,1,9), variando p y q. Este proceso puede durar algunas horas ya que realiza 81 modelos ARIMA para encontrar el mejor.

El grado de diferenciación d se mantiene en 1, ya que se estima que los datos son estacionarios en primeras diferencias.

Al realizar el modelo se obtiene una matriz de los AIC encontrados tal como la que se observa en la tabla 2.5.2.

Tabla 2.5.3: Ejemplo de una matriz de criterios de información de Akaike.

	q1	q2	q3	q4	q5	q6	q7	q8	q9
p1	77044,4	77043,7	77030,1	77030,1	77028,3	77015,2	77015,2	77015,2	77008,7
p2	77044,5	77046,6	77032,0	77017,3	77020,0	77020,0	77020,0	77017,3	77017,3
p3	77030,7	77041,3	77046,7	77046,7	77017,3	77005,6	77011,9	77002,3	77002,3
p4	77025,5	77027,5	77027,5	77003,7	77000,4	77000,4	77000,4	77002,3	77002,3
p5	77027,4	77019,7	77020,5	77003,7	77003,0	77003,0	77017,3	77017,3	77017,3
p6	77012,7	77012,7	77020,5	77003,7	77000,2	77000,4	77009,5	77017,3	77017,3
p7	77014,4	77016,5	77007,2	77009,5	76998,4	77002,9	77009,5	77017,3	77017,3
p8	77061,3	77013,4	77011,4	77000,4	77002,3	76995,4	76990,1	76990,1	76990,1
p9	77062,5	77009,4	77003,4	77002,1	76986,8	76993,6	76992,6	76993,6	76977,4

Para el caso de la tabla 2.5.3, el mejor modelo resulta ser un modelo ARIMA(9,1,9), mientras que la función auto.arima() señalaba un modelo ARIMA(1,1,2).

2.5.2.4. Modelo combinado: Como ya se ha mencionado, el modelo ARIMA supone que la serie es estacionaria en diferencias, pero pozos con régimen de flujo pseudoestable, cuya tendencia se mantenga, pueden ofrecer datos estacionarios en tendencia, por tanto, en este modelo se calcula el mejor modelo hiperbólico (que representa la componente tendencial) y se calcula un modelo ARIMA con la función auto.arima para los residuos (que representa las otras componentes de la serie de tiempo).

2.5.3. Realización del pronóstico: La metodología ARIMA (componente principal del software realizado) debe usarse para hacer pronósticos a corto plazo (alrededor de 30 días o intervalos de tiempo).

Si se tienen más de cinco años de datos de producción es aconsejable utilizar el método combinado o el hiperbólico automático para realizar los pronósticos.

En caso de no tener muchos datos de producción (tiempos tempranos), se aconseja realizar un modelo hiperbólico con el exponente n sugerido por Fetkovich⁵⁷ para los distintos mecanismos de empuje (ver sección 1.1 y tabla 1.1.1).

⁵⁷ FETKOVICH, M.J., FETKOVICH, E.J., & FETKOVICH, M.D. Useful Concepts for Decline-Curve Forecasting, Reserve Estimation, and Analysis. Paper SPE 28628. 1994

El pronóstico para las curvas de Arps es una extrapolación programada directamente en c#. En cuanto a ARIMA se realiza utilizando funciones de R con la ayuda de R.Net, pero se deben tener en cuenta algunas recomendaciones:

- Funciones: Las dos funciones que permiten hacer pronósticos con un modelo ARIMA ya construido en R son: `forecast()` y `predict()`, que reciben como parámetros el modelo (la variable que se uso para guardar el modelo, no una cadena de caracteres), y el número de intervalos de tiempo a pronosticar.
- Pronosticar modelos hechos con `arima()`: Los modelos realizados con la función `arima()` pueden ser utilizados por la función `forecast` en la consola de R, pero al ser utilizado por R.Net, lanza una excepción que impide el correcto funcionamiento, por tanto, debe ser utilizada la función `predict()`.
- Modelos hechos con `auto.arima()`: Sucede lo contrario que con la función `arima()`. Estos modelos si pueden ser utilizados por la función `forecast` para realizar el proceso adecuadamente, pero al utilizar la función `predict()` se obtiene una excepción que rompe el proceso e impide la correcta ejecución del software.
- Intervalo de confianza: La función `forecast()` además de realizar los pronósticos, también genera intervalos de confianza del 80% y el 95% para la predicción.

3. MANUAL DE USUARIO

DeclineR 1.0 fue desarrollado con Visual Studio Community 2015 haciendo uso de la licencia gratuita para programadores individuales. Para más información visite la página web <https://www.visualstudio.com/license-terms/mt171547/>.

DeclineR 1.0 hace uso de software libre como:

- R, software libre bajo licencia GNU General Public Licence⁵⁸ y sus respectivos términos de Copyleft.

Para más información sobre R se recomienda visitar su página oficial <https://www.R-project.org/>.

⁵⁸ R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Se aclara que DeclineR no hace modificaciones ni distribuye copias de dicho software.

- R.Net, paquete para Visual studio. Respetando los términos de uso de RDotNet se reproduce el siguiente aviso legal:

“Copyright (c) 2010, RecycleBin

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”⁵⁹

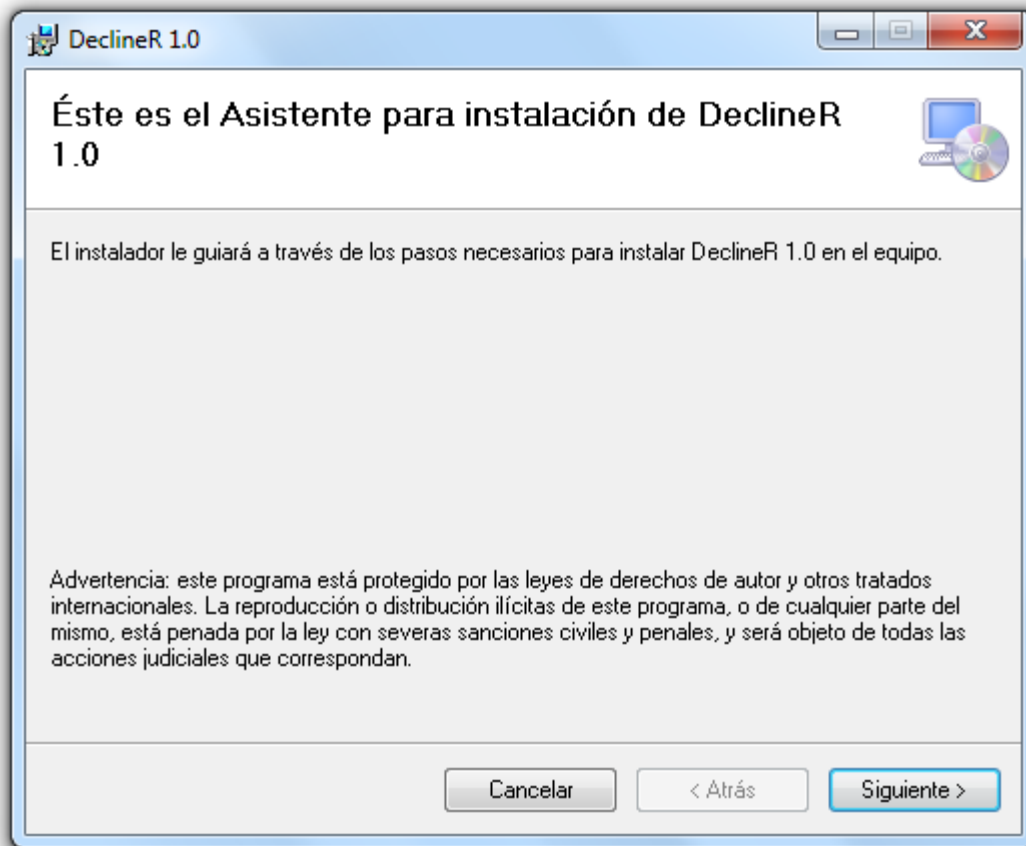
Para más información sobre R.Net y sus condiciones de uso visite el sitio web <https://rdotnet.codeplex.com>.

3.1. INSTALACIÓN Y EJECUCIÓN DEL SOFTWARE

3.1.1. Instalación

⁵⁹ R.Net. Paquete de interoperabilidad entre visual studio y R. Términos y condiciones de uso. Revisado 13 de enero de 2017. Disponible en: <https://rdotnet.codeplex.com/license>.

Figura 3.1.1. Asistente para la instalación de DeclineR 1.0.

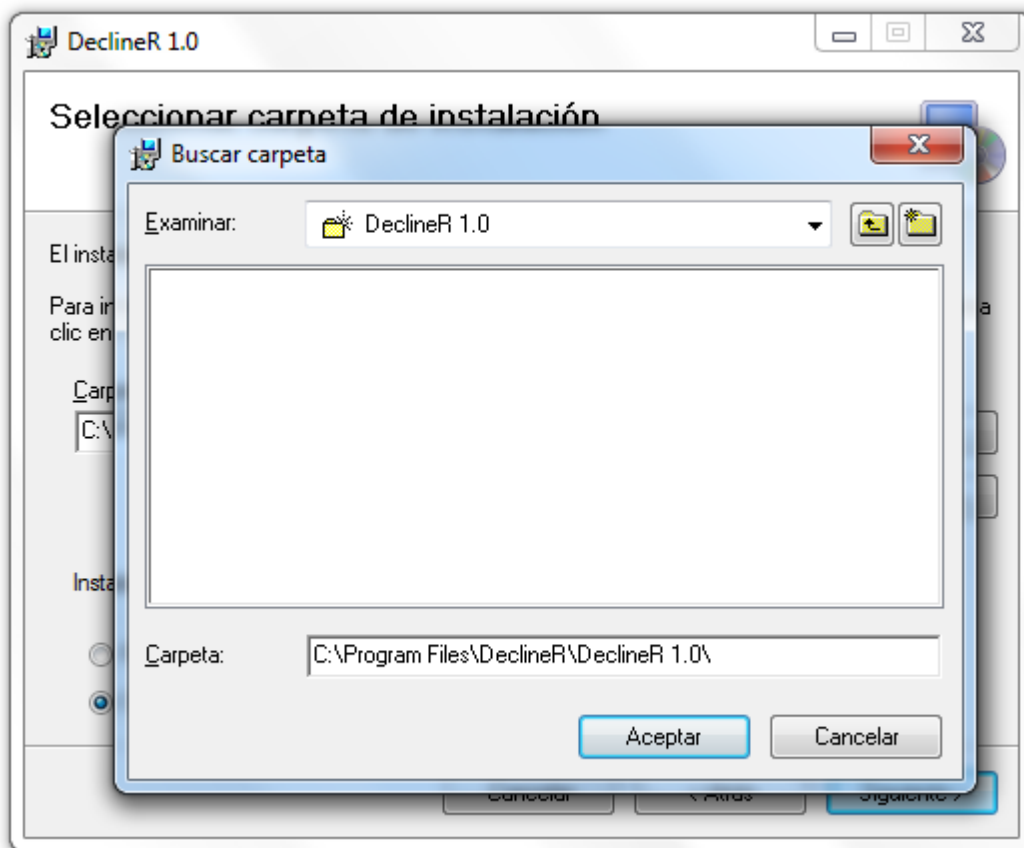


Para la instalación se requiere:

- Windows vista / 7 / 8 / 10.
- 3 Mb de espacio en disco disponible para "DeclineR 1.0".
- Tener instalado R (versión de 32 bits) o tener 150 Mb de espacio en disco para instalarlo.

Abra el cd de instalación del software y abra el archivo “setup.exe”, se abrirá enseguida el asistente de instalación (ver Figura 3.1.1).

Figura 3.1.2: elección de la carpeta de instalación.



Haga clic en “Siguiente”, se le mostrará la ruta de instalación predeterminada “C:\Program Files\DeclineR\DeclineR 1.0\”, si desea instalarlo en esta ruta haga clic en “Siguiente”, de lo contrario haga clic en el botón “Examinar” y elija la carpeta deseada (ver Figura 3.1.2).

Haga clic en “Siguiente” para confirmar la instalación, Si aparece una ventana emergente de confirmación, otorgue los permisos necesarios para realizar cambios en el equipo.

3.1.2. Ejecutar DeclineR 1.0 por primera vez

Figura 3.1.3: Pantalla de inicio:

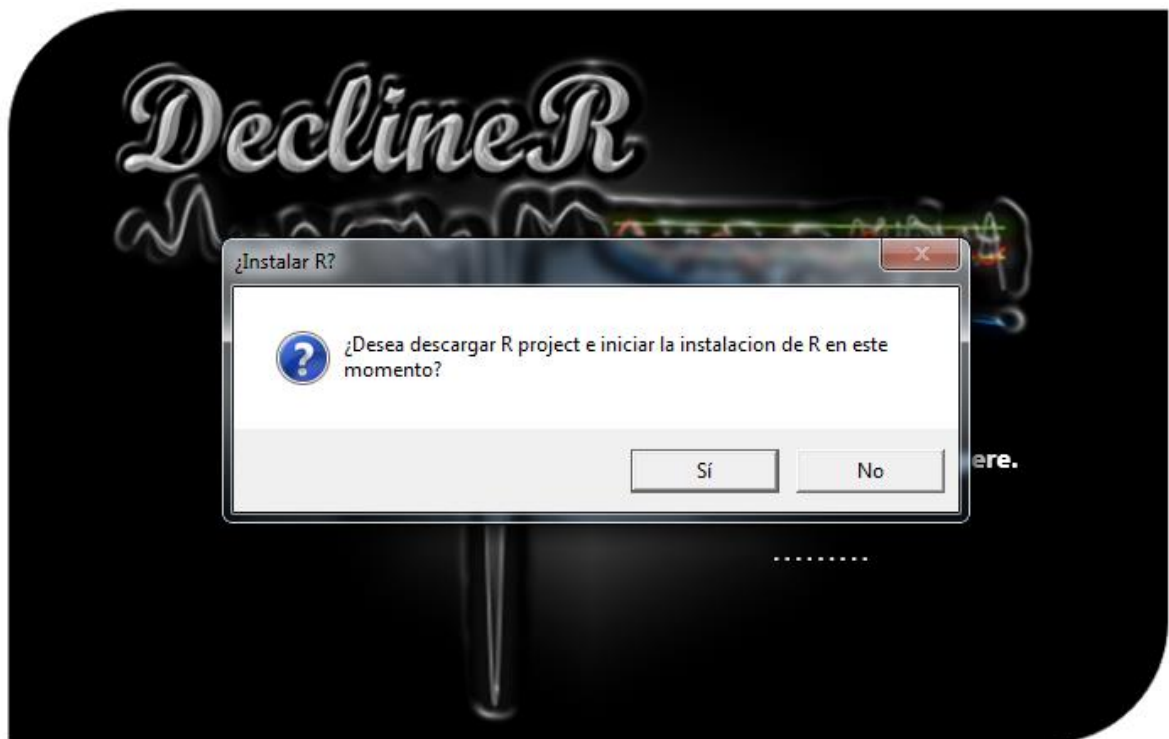


Para la correcta ejecución de DeclineR 1.0 es necesario tener instalado R Project con las librerías “forecast” y “TSA”, o tener una conexión a internet estable para permitir la descarga e instalación de R Project.

Ejecute el programa, observará una pantalla de inicio (ver Figura 3.1.3), haga clic en “Iniciar” o pulse la tecla “Enter” para continuar.

Si R Project no está instalado aparecerá un mensaje indicándole que lo instale para poder continuar, haga clic en “Aceptar”, Inmediatamente verá otro mensaje que ofrece descargar e instalar R (ver Figura 3.1.4), Si hace clic en “No”, se le ofrecerá visitar la página oficial de R Project (<https://www.R-project.org/>).

Figura 3.1.4: Opción de instalación automática de R.



Si hace clic en “Si” la descarga de R se iniciará internamente (Ver Figura 3.1.5). Por favor no cierre el programa mientras se realiza la operación. La descarga puede durar uno o varios minutos dependiendo de la calidad de la conexión. Una vez descargado, se abrirá el asistente de instalación de R Project (ver Figura 3.1.6) y se cerrará DeclineRing 1.0.

Figura 3.1.5: Descarga interna de R.

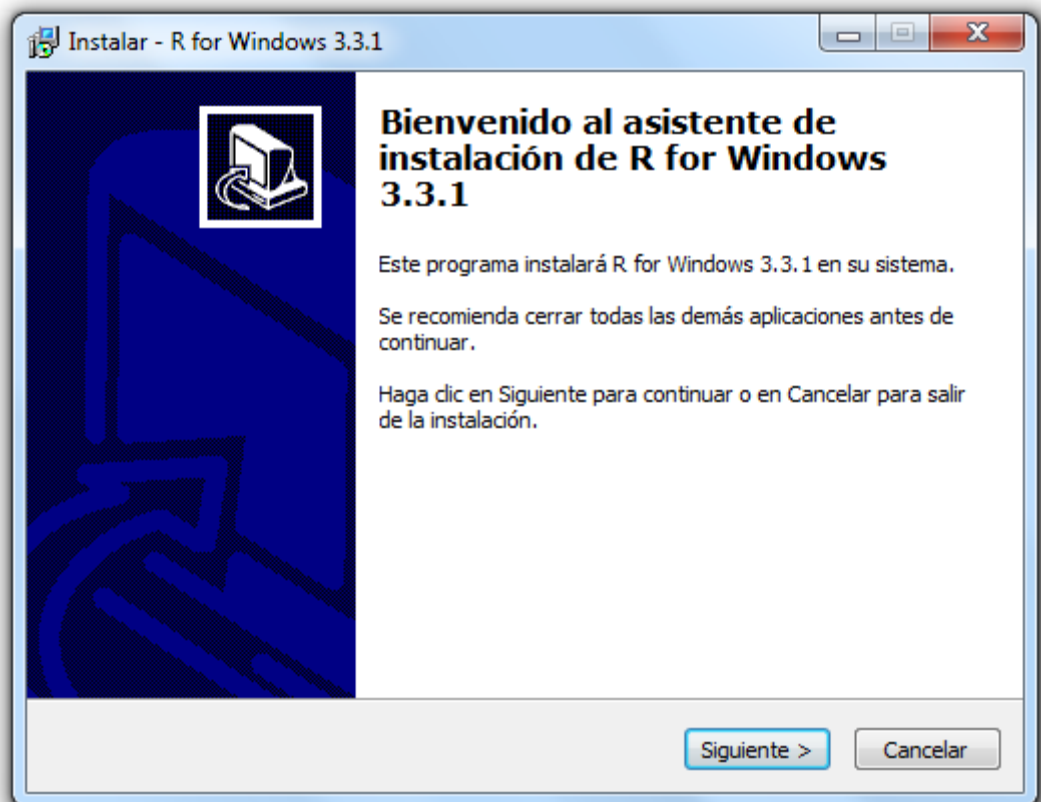


Una vez abierto el asistente para la instalación de R, siga los pasos correspondientes, lea los términos y condiciones, y continúe hasta instalar correctamente R. En la sección de opciones de configuración, la instalación más

sencilla se realiza marcando la opción “No” para que se instale la configuración predeterminada.

Una vez finalizada la instalación de R, abra nuevamente DeclineR 1.0, esta vez se mostrarán mensajes para instalar las librerías necesarias, Haga clic en “Si” cada vez que sea necesario.

Figura 3.1.6: Asistente para la instalación de R.⁶⁰



⁶⁰ R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

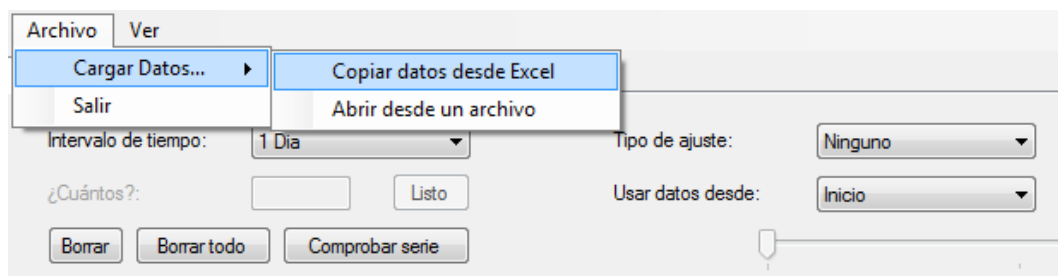
Una vez instaladas las librerías, DeclineR 1.0 estará listo para ser utilizado.

3.2. CARGA DE DATOS

Se permite la copia de columnas desde Excel, o carga directa de archivos .txt y .csv.

3.2.1. Copia de datos desde Excel: Para copiar columnas de datos siga la ruta Archivo/Cargar Datos.../Copiar datos desde Excel (Ver Figura 3.2.1).

Figura 3.2.1: Ruta para copiar datos.



Se abrirá enseguida un formulario (ver Figura 3.2.2). Si tiene los datos con sus respectivas fechas, copie las columnas según corresponda.

En caso de no tener las fechas directamente relacionadas en otra columna, seleccione “Definir rango de fechas”, y proceda a definir las fechas inicial y final del rango para poder añadirlo (ver Figura 3.2.3). Puesto que es necesario que los datos ingresados correspondan a una serie de tiempo, se añadirá solo un rango de fechas, y cada rango nuevo sustituirá al anterior. Una vez añadido el rango, seleccione la primera celda de la columna de datos, y pulse Ctrl + v, o el botón Pegar.

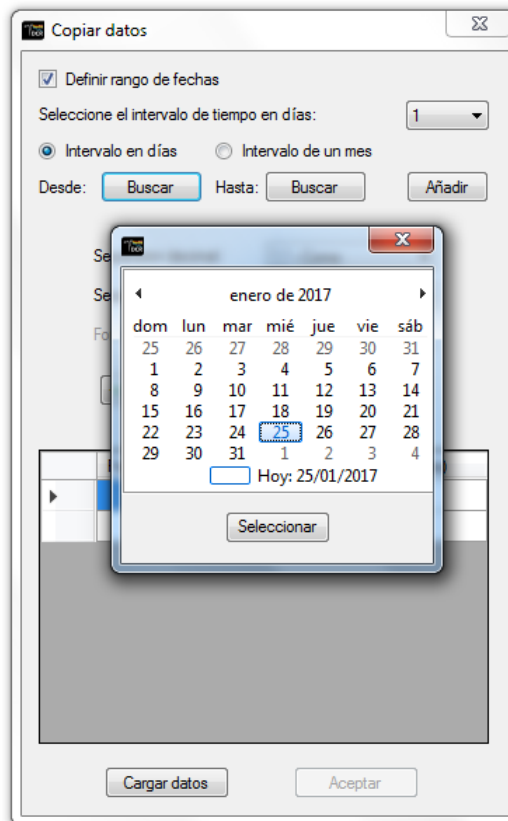
Figura 3.2.2: Formulario para copiar datos.

Fecha	Produccion (STB/dia)

Una vez copiados los datos, debe seleccionar el separador decimal, el separador de miles (en caso de tenerlo) y el formato de fecha, para poder importar los datos y manejarlos correctamente.

Cuando este seguro de que ha ingresado correctamente los datos, haga clic en el botón “Cargar datos”, y por último en el botón “Aceptar”.

Figura 3.2.3: Selección de fechas.



Nota: El botón “Borrar datos” borrará todos los datos ingresados. Los botones “+” y menos estarán disponibles para añadir filas en blanco, y eliminar renglones de datos en cualquier posición.

3.2.2. Carga directa de archivos: Para cargar archivos .csv o .txt siga la ruta Archivo/Cargar Datos.../Abrir desde un archivo (Ver Figura 3.2.5).

Se abrirá enseguida un formulario que permite cargar datos desde archivos (Ver Figura 3.2.4).

Figura 3.2.4: Formulario para cargar archivos.

Cargar Archivo

Buscar archivo No se ha seleccionado ningún archivo

Separador columnas: Tabulación

Títulos contenidos en el archivo

Visualizar

Columna de fechas:

Columna de datos:

Separador decimal: " - Coma

Separador de miles: Ninguno

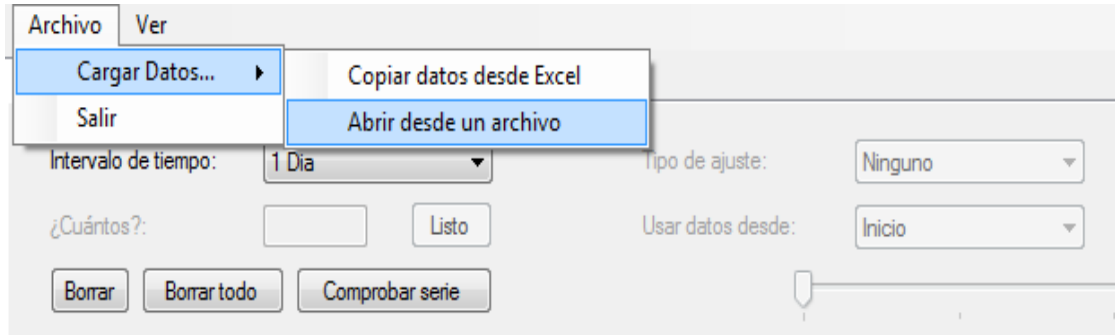
Formato de fecha: dd/MM/yyyy

+ - Borrar datos

1

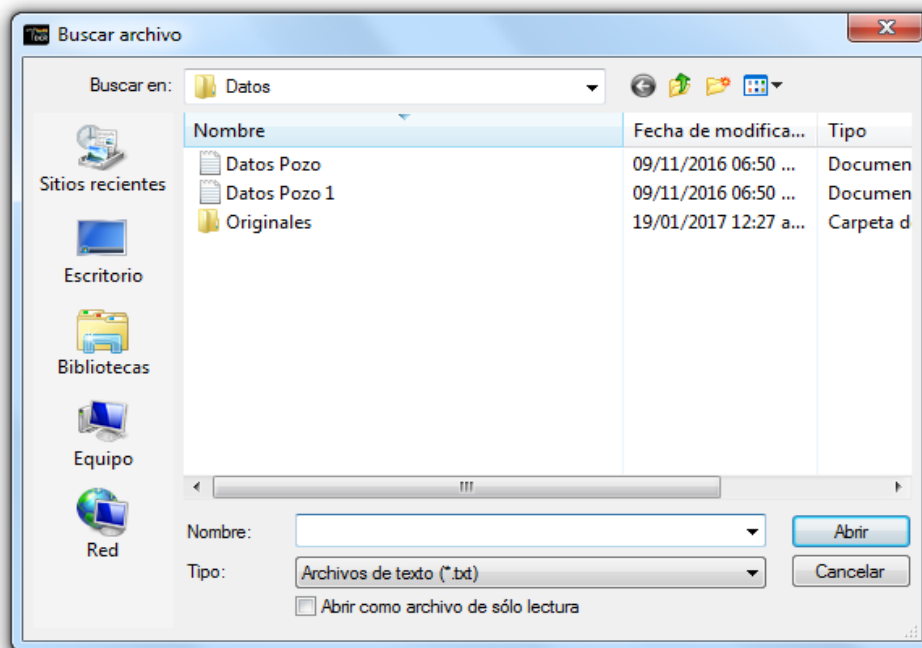
Cargar datos Aceptar

Figura 3.2.5: Abrir un archivo de datos.



Haga clic en el botón buscar archivo, se abrirá un cuadro de diálogo explorador, que le permitirá abrir el archivo deseado (Ver Figura 3.2.6). Asegúrese de seleccionar el tipo de archivo correcto en la sección “tipo” en la parte inferior del cuadro de diálogo.

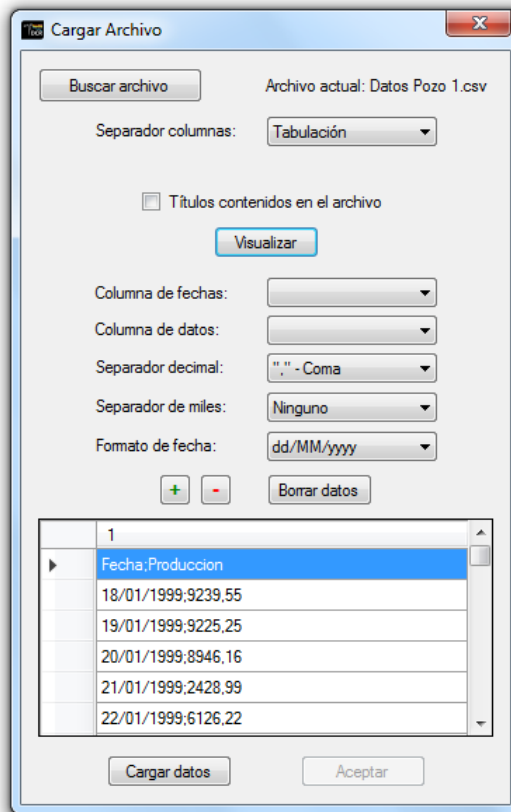
Figura 3.2.6: Cuadro de diálogo “Buscar Archivo”.



Una vez seleccionado el archivo, haga clic en el botón “Visualizar”, observará que se despliega la información contenida en el archivo (ver Figura 3.2.7).

Identifique si hay títulos contenidos en la primera fila, de ser así marque la casilla “Títulos contenidos en el archivo”, elimine los espacios vacíos con datos no relevantes en las primeras filas con el botón “-”, y seleccione el separador de columnas adecuado (en el caso de la imagen es punto y coma “;”). Una vez hecho esto, haga clic nuevamente en el botón “Visualizar”, deberá observar los datos organizados en sus respectivas columnas (ver Figura 3.2.8).

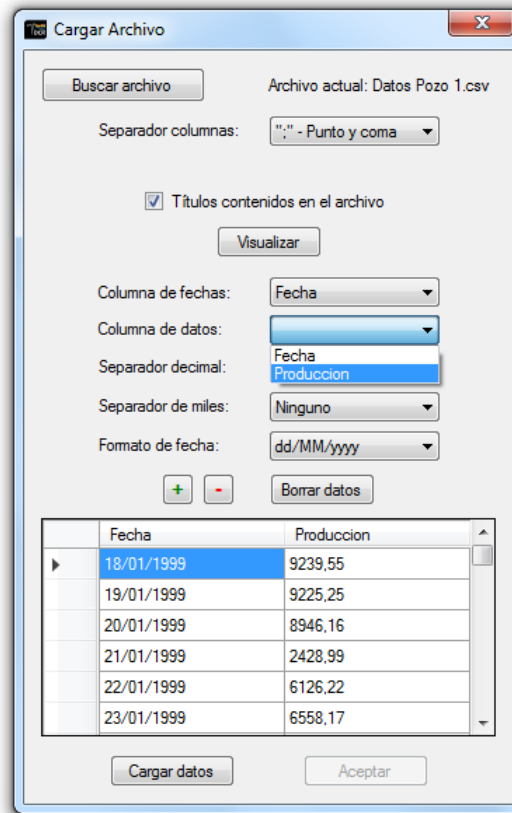
Figura 3.2.7: Primera vista del archivo cargado.



Proceda a seleccionar la columna que contiene las fechas, la columna que contiene los datos (ver Figura 3.2.8), el separador decimal, el separador de miles (en caso de tenerlo) y el formato de fecha, para poder importar los datos y manejarlos correctamente.

Cuando este seguro de que ha ingresado correctamente los datos, haga clic en el botón "Cargar datos", y por último en el botón "Aceptar".

Figura 3.2.8: Datos organizados y listos para cargar.



3.3. MANEJO DE DATOS

En la sección "Intervalo de tiempo" elija el intervalo de tiempo que desea manejar de acuerdo con los datos. Una vez elegido haga clic en el botón "Comprobar serie". Los datos se graficarán automáticamente.

3.3.1. Ajuste de los datos:

Se proporcionan las siguientes opciones de ajuste:

- Ajuste de media móvil: Reemplaza los valores cero (datos faltantes de producción, llamados datos atípicos) con el promedio aritmético de los siete datos inmediatamente anteriores.
- Ninguno: No se realiza ningún ajuste, se incluirán los datos atípicos (datos de cero producción) en la estimación del modelo. Esto puede disminuir la capacidad predictiva del modelo.

3.3.2. Selección de datos: En la parte superior se encuentran una serie de controles que le permitirán seleccionar gráficamente los datos que van a ser usados para realizar el modelo (ver Figura 3.3.1 y Figura 3.3.2).

3.3.2.1. Usar datos desde: Si desea realizar un modelo de Arps, debe recordar que una de las suposiciones es el estado pseudoestable, por tanto, debe ignorar los datos que corresponden a flujo radial.

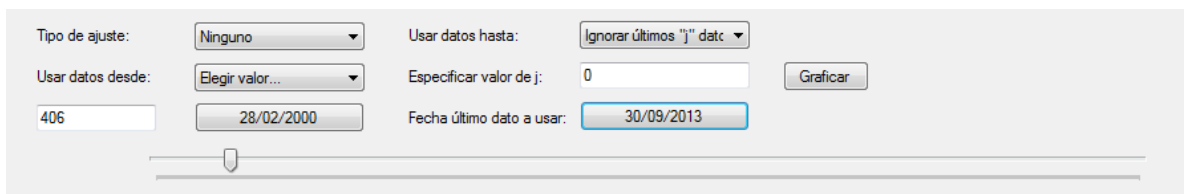
Otra suposición que se emplea en los modelos de Arps es que el pozo no se estimula en el transcurso del tiempo que se está pronosticando, por tanto, es conveniente seleccionar los datos posteriores al último cambio de declinación inducido.

Se ofrecen tres posibilidades:

- Inicio: Se usarán todos los datos para realizar el modelo.

- Punto máximo: Selecciona los datos a partir del punto máximo de producción, suponiendo que éste marca el inicio de la declinación y por tanto del estado pseudoestable.
- Elegir valor: Habilita la barra de desplazamiento (ver Figura 3.3.1) para permitir seleccionar manualmente el punto desde el cual se van a utilizar los datos. Para facilitar la selección de los datos, habrá un marcador púrpura desplazándose con la barra mencionada (ver Figura 3.3.2). También es posible seleccionar directamente la fecha desde la cual se van a usar los datos haciendo clic en los botones que muestran la fecha que se ha seleccionado, o digitando un número en el cuadro de texto ubicado bajo la etiqueta “Usar datos desde”. El número que se muestra corresponde a la posición del dato seleccionado en la lista que se ingresó.

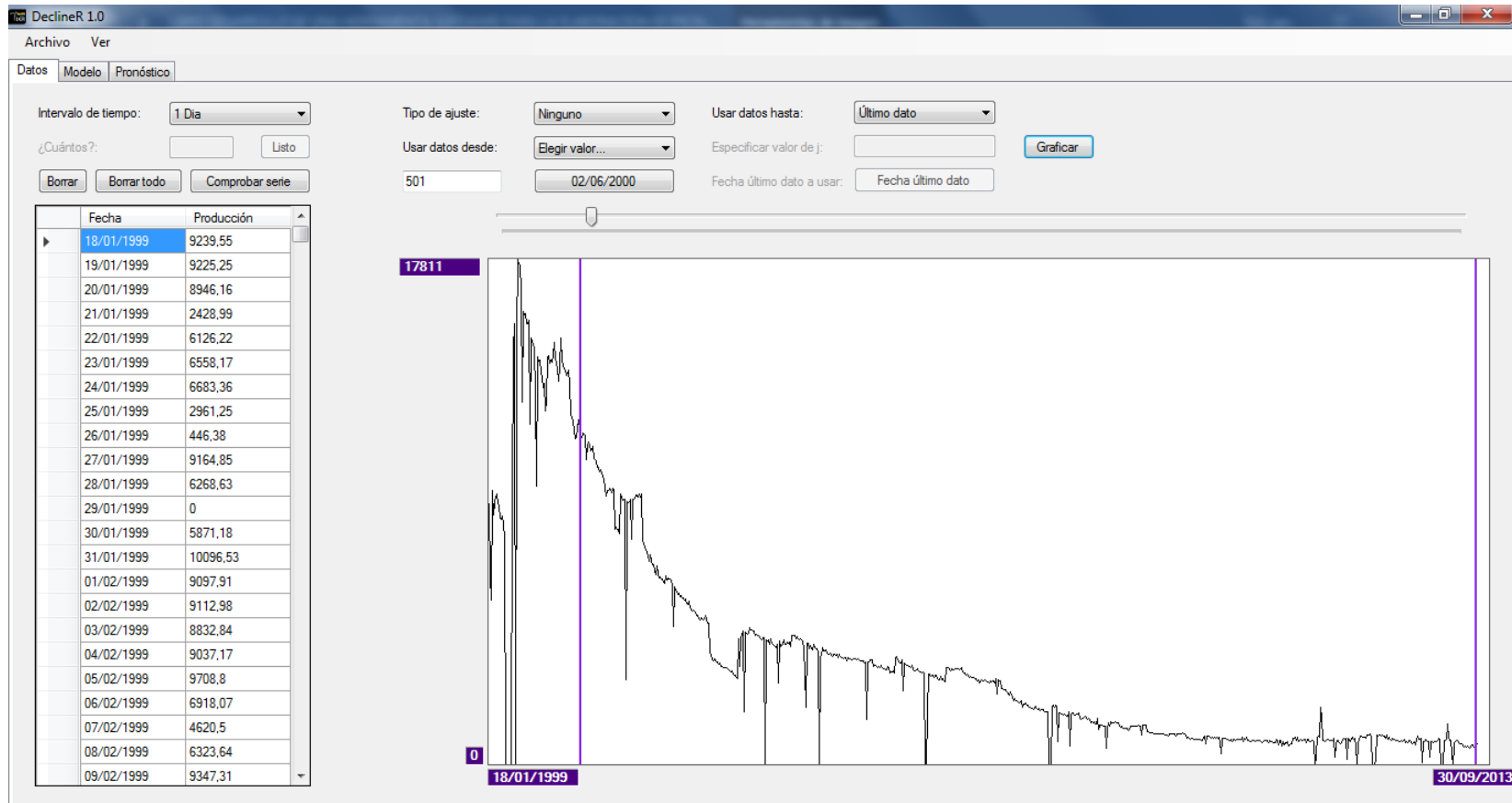
Figura 3.3.1: Controles para selección de datos.



The image shows a user interface for data selection. It includes several input fields and a slider. The fields are: 'Tipo de ajuste' (set to 'Ninguno'), 'Usar datos hasta' (set to 'Ignorar últimos "j" datos'), 'Usar datos desde' (set to 'Elegir valor...'), 'Especificar valor de j' (set to '0'), and 'Fecha último dato a usar' (set to '30/09/2013'). There is also a 'Graficar' button. Below these fields is a horizontal slider with a purple marker.

Tipo de ajuste:	Ninguno	Usar datos hasta:	Ignorar últimos "j" datos	
Usar datos desde:	Elegir valor...	Especificar valor de j:	0	Graficar
406	28/02/2000	Fecha último dato a usar:	30/09/2013	

Figura 3.3.2: Página de manejo de datos



3.3.2.2. Usar datos hasta: Los datos finales suelen ser los que más importancia tienen al momento de realizar un pronóstico. Asegúrese de que los datos finales no sean datos atípicos o irregulares.

Se ofrecen las siguientes opciones:

- **Último dato:** Se usarán los datos desde el punto especificado (Ver sección 3.3.2.1.) hasta el último dato ingresado.
- **Ignorar últimos “j” datos:** Habilita un cuadro de texto y un botón para ingresar el número de datos que se desea ignorar (el número debe ser un entero positivo) o la fecha hasta la cual desea utilizar los datos.

Evite ignorar un número demasiado grande de datos, ya que esto disminuye la capacidad predictiva del modelo.

Una vez seleccionados y ajustados los datos continúe a la página “Modelo”.

Nota: Puede regresar en cualquier momento a la página “Datos” para realizar las modificaciones pertinentes.

3.4. CONSTRUCCIÓN Y ELECCIÓN DEL MODELO

Podrá revisar en cualquier momento la información de los modelos que ha utilizado siguiendo la ruta Ver/Ver información de modelos usados.

Se ofrecen cuatro principales opciones para la construcción de modelos:

- Exponencial (Arps)
- Armónico (Arps)
- Hiperbólico (Arps)
- ARIMA (Box & Jenkins)

3.4.1. Construcción de un modelo Exponencial (Arps):

Hay tres posibilidades para la construcción del modelo (ver Fig 3.4.1):

- Especificar un valor de declinación inicial: Se construye el modelo con la declinación que se especifique en el cuadro de texto.
- Encontrar modelo que pase por el punto final: Se soluciona la curva exponencial de Arps obligando al modelo a pasar por el punto de producción inicial definido en la página “Datos” como primer dato (ver sección 3.3.2.1.) y el último dato de producción seleccionado (ver sección 3.3.2.2.).
- Encontrar modelo con mínimos cuadrados: Se realiza la regresión con la matriz de mínimos cuadrados para encontrar los valores de los parámetros del modelo (producción inicial y declinación inicial).

Seleccione la opción deseada y haga clic en el botón calcular. Se graficará inmediatamente el modelo obtenido y se desplegarán los resultados de parámetros y criterios de selección en el panel de resultados.

Figura 3.4.1 Opciones para la construcción de un modelo exponencial.

Modelo Exponencial (Arps) | Modelo Armónico (Arps) | Modelo Hiperbólico (Arps) | Modelo ARIMA

Especificar valor de Di: Declinación Inicial:

Encontrar modelo que pase por el punto final

Encontrar el modelo con mínimos cuadrados

Calcular

3.4.2. Construcción de un modelo Armónico (Arps):

Hay tres posibilidades para la construcción del modelo (ver Figura 3.4.2.):

- Especificar un valor de declinación inicial: Se construye el modelo con la declinación que se especifique en el cuadro de texto.
- Encontrar modelo que pase por el punto final: Se soluciona la curva armónica de Arps obligando al modelo a pasar por el punto de producción inicial definido en la página “Datos” como primer dato (ver sección 3.3.2.1.) y el último dato de producción seleccionado (ver sección 3.3.2.2.).
- Encontrar modelo con mínimos cuadrados: Se realiza la regresión con la matriz de mínimos cuadrados para encontrar los valores de los parámetros del modelo (producción inicial y declinación inicial).

Seleccione la opción deseada y haga clic en el botón calcular. Se graficará inmediatamente el modelo obtenido y se desplegarán los resultados de parámetros y criterios de selección en el panel de resultados.

Figura 3.4.2: Opciones para la construcción de un modelo armónico

Modelo Exponencial (Arps) | Modelo Armónico (Arps) | Modelo Hiperbólico (Arps) | Modelo ARIMA

Especificar valor de Di: Declinación Inicial:

Encontrar modelo que pase por el punto final

Encontrar el modelo con mínimos cuadrados

Calcular

Figura 3.4.3: Controles para un modelo hiperbólico de Arps.

Modelo Exponencial (Arps) | Modelo Armónico (Arps) | Modelo Hiperbólico (Arps) | Modelo ARIMA

Especificar Declinación inicial y exponente n Declinación Inicial: 0,001

Obligar al modelo a pasar por el punto final Exponente n:

Iterar para encontrar el mejor modelo

Numero de iteraciones: 20

Permitir exponente n atípico

Variar n desde: Variar n hasta:

Calcular

3.4.3. Construcción de un modelo Hiperbólico (Arps): Para la construcción del modelo hiperbólico de Arps, se plantean tres opciones, una totalmente manual, otra semiautomática y una totalmente automática (ver Figura 3.4.3):

- Especificar declinación inicial y exponente n : Habilita el cuadro de texto para especificar la declinación inicial (recuerde que debe ser un número decimal positivo), y una barra deslizable que permite elegir el exponente n (ver Figura 3.4.3) y cambiarlo, observando el ajuste que presenta y los resultados tanto de parámetros como de criterios de selección en el panel de resultados.
- Obligar al modelo a pasar por el punto final: Esta es la opción semiautomática. Habilita únicamente la barra deslizable para elegir el valor del exponente n (ver Figura 3.4.4), la declinación inicial se calcula internamente.
La gráfica irá variando según se vaya desplazando la barra, para garantizar que el ajuste visual y los resultados se puedan observar en tiempo real.
- Iterar para encontrar el mejor modelo: Construye automáticamente el modelo que mejor se ajusta a los datos y proporciona los resultados correspondientes. Habilita la opción de cambiar el número de iteraciones que se realicen (ver Figura 3.4.5).
En las iteraciones se selecciona el mejor exponente n encontrado, y se permiten exponentes n atípicos (menores que cero y mayores que uno) con el fin de garantizar un buen ajuste, debido a esto, como resultado puede obtenerse un modelo exponencial, armónico o hiperbólico, dependiendo del ajuste que se presente.

En las opciones manual y semiautomática es posible habilitar el uso de exponentes n atípicos marcando la casilla “Permitir exponente n atípico”, y eligiendo los valores límite.

En la opción automática, el modelo resultante además de ser el mejor modelo hiperbólico, es el mejor modelo de Arps que pasa por los puntos inicial y final.

Tenga en cuenta que puede haber combinaciones de declinación inicial y exponente n (principalmente atípicos) que no converjan, en ese caso se enviará un mensaje de error indicando que debe cambiar dichos valores (ver Figura 3.4.4).

Figura 3.4.4: Mensaje de error en la construcción del modelo.

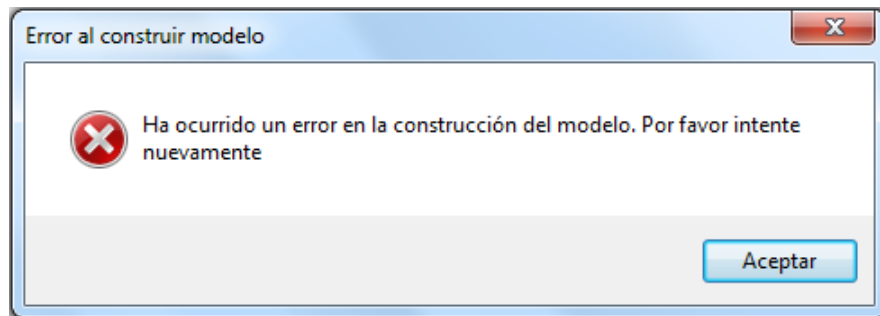
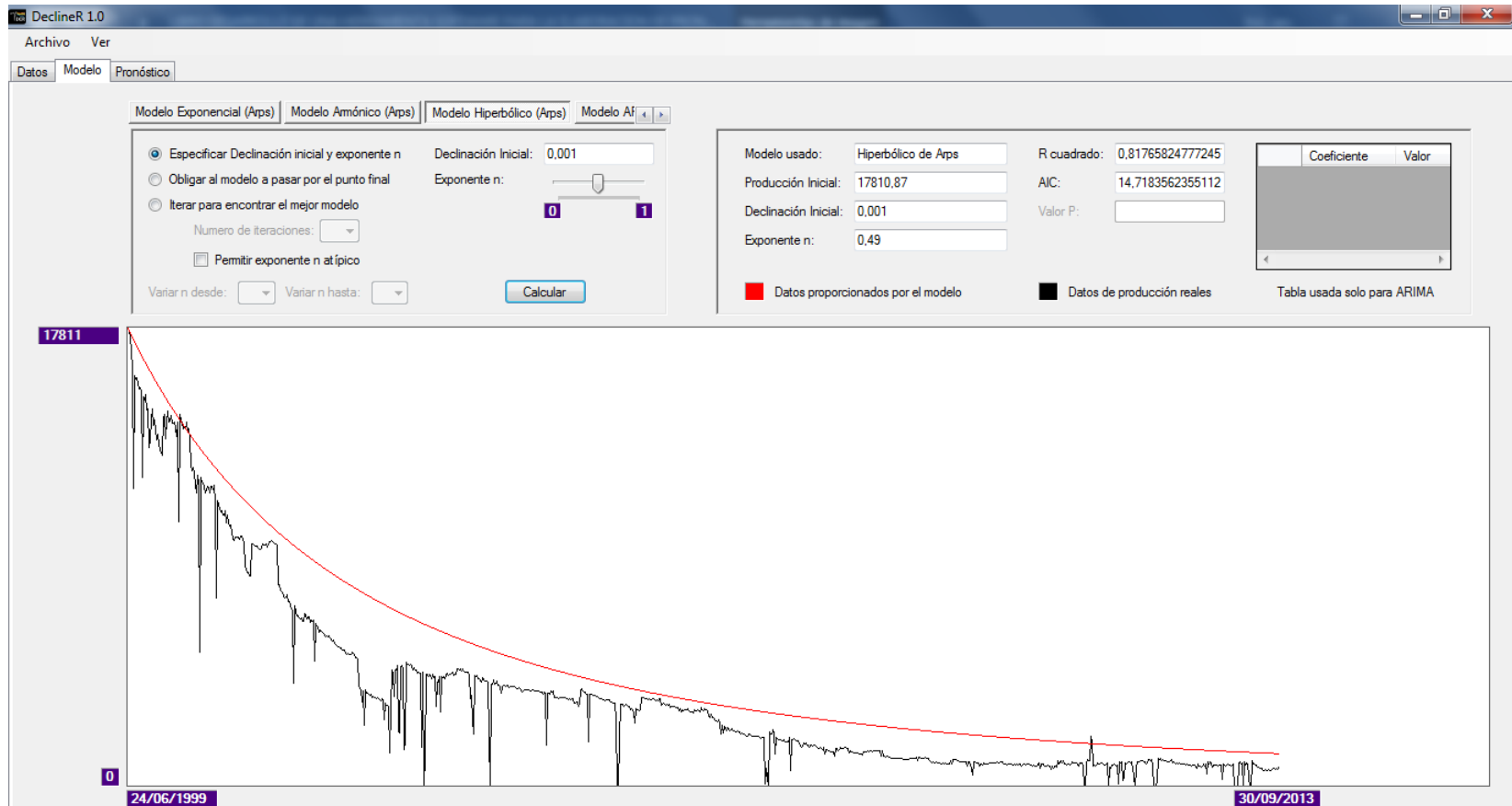


Figura 3.4.5: Modelo Hiperbólico especificando D_i y n (opción totalmente manual).



Nota: En la parte superior izquierda se observa el panel de controles del modelo y en la superior derecha el panel de resultados.

Figura 3.4.6: Modelo Hiperbólico especificando n (opción semiautomática).

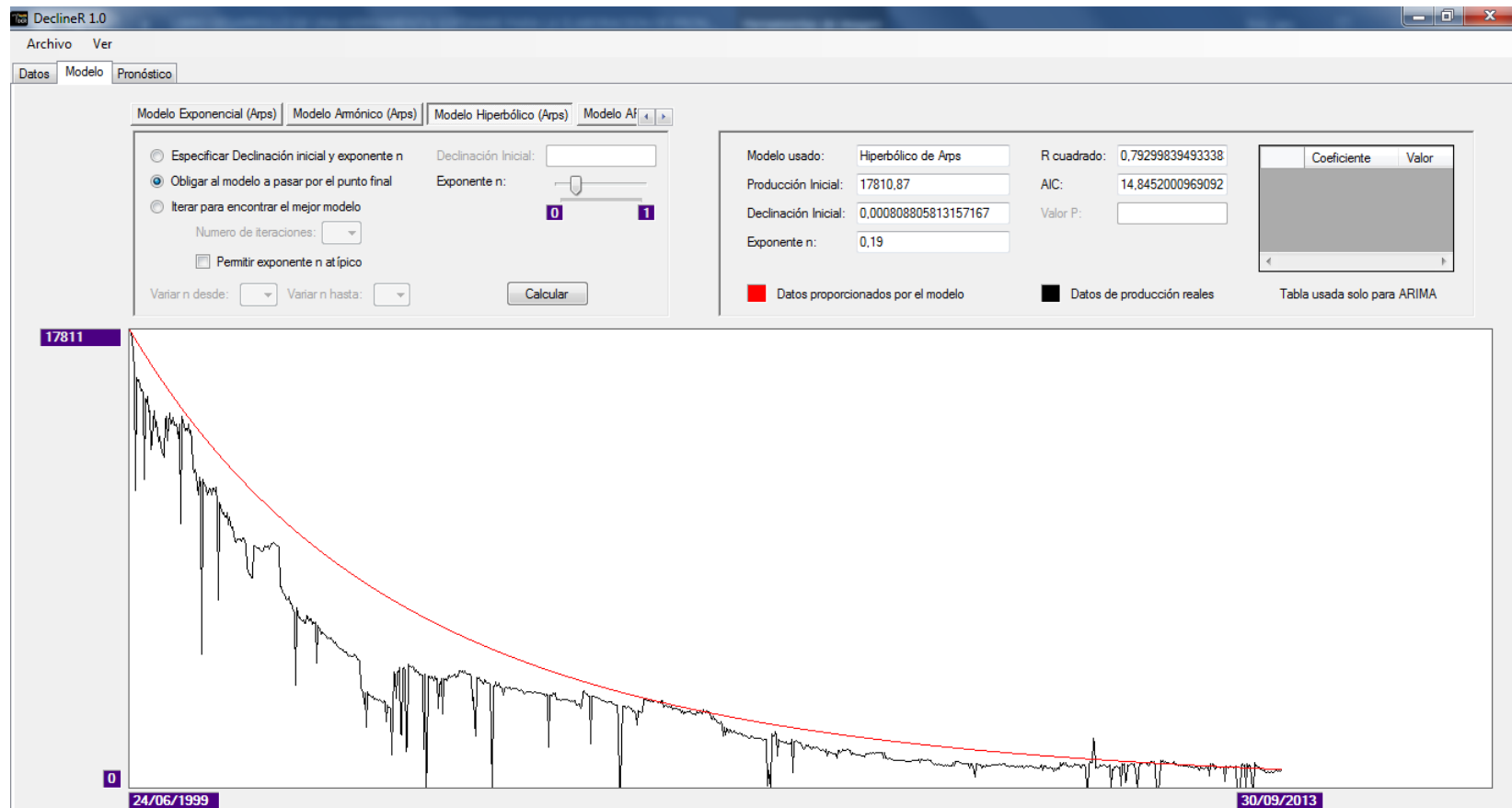
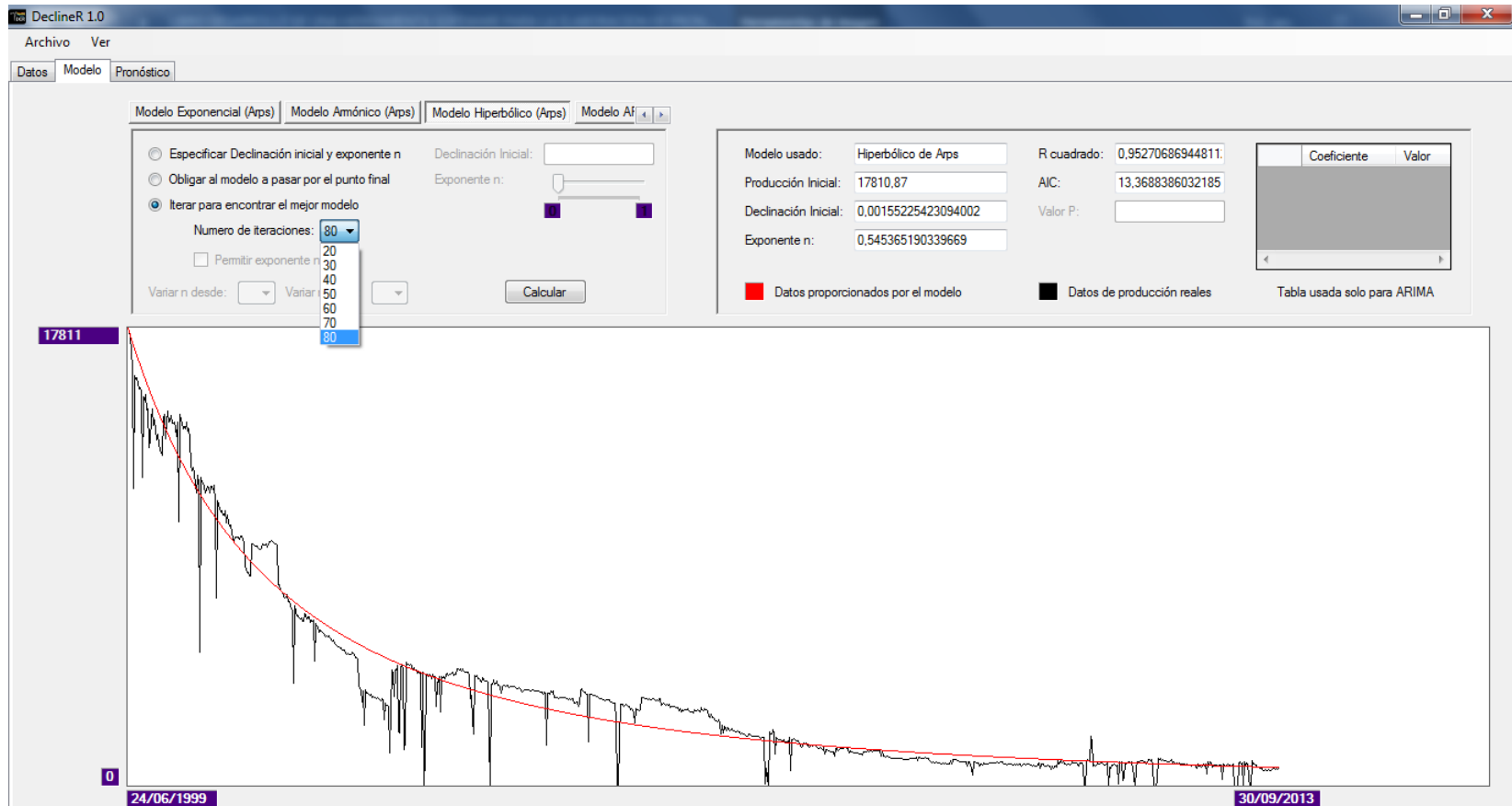


Figura 3.4.7: Modelo Hiperbólico automático.



3.4.4. Construcción de un modelo ARIMA (Box & Jenkins): El modelo ARIMA es realizado utilizando el software y lenguaje de programación R⁶¹, y se ofrecen tres opciones:

- Usar algoritmo auto.arima de R⁶²: La función devuelve el mejor modelo arima de acuerdo al valor del AIC, AICc o BIC.
- Especificar el orden del modelo ARIMA (p,d,q): Habilita tres cuadros de texto para especificar el orden del modelo deseado (“Ver Figura 3.4.8”). Los valores de orden deben ser enteros positivos. Una vez seleccionado el orden, haga clic en el botón “Calcular”.
- Hallar modelo con mejor AIC: Realiza todos los modelos ARIMA(p,d,q) variando p y q entre 1 y 9, por lo tanto se obtienen 81 modelos entre los cuales se elige el que tenga mejor criterio de información de Akaike. Este procedimiento puede tardar horas.

Como apoyo para la selección del orden del modelo, se da la opción de realizar la función de autocorrelación simple (ver Figura 3.4.6), FAS, y la función de autocorrelación parcial (ver Figura 3.4.7), FAP, de los datos del modelo. Estas funciones también son realizadas utilizando R (funciones `acf()` y `pacf()`)⁶³.

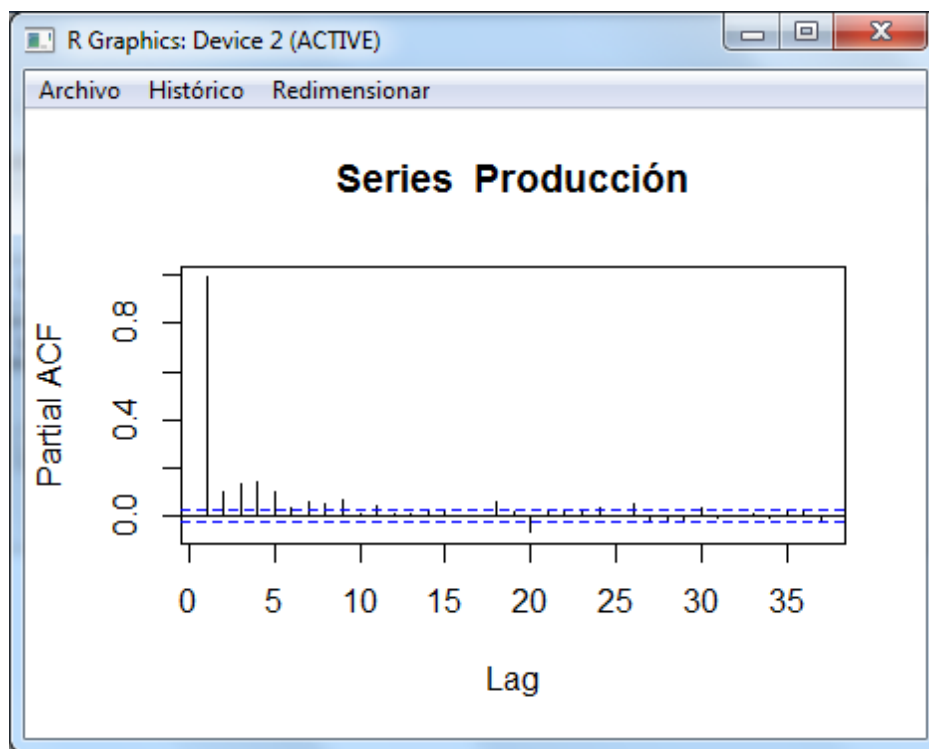
⁶¹ R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

⁶² Hyndman, R.J. and Khandakar, Y. (2008) "Automatic time series forecasting: The forecast package for R", *Journal of Statistical Software*, **26**(3).

⁶³ Original: Paul Gilbert, Martyn Plummer. Extensive modifications and univariate case of `pacf` by B. D. Ripley. For R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

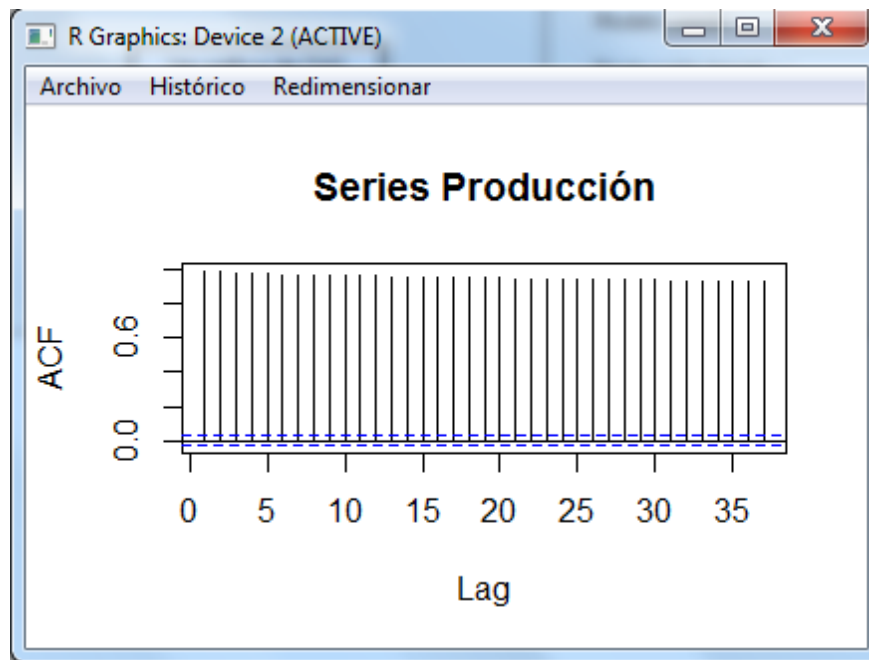
Para realizar las funciones de autocorrelación simple y parcial siga en cualquier momento la ruta ver/Función de Autocorrelación Simple y ver/Función de Autocorrelación Parcial, o directamente desde el panel de modelo ARIMA haciendo clic en los botones “Ver gráfica de FAS” y “Ver gráfica de FAP” respectivamente.

Figura 3.4.8: Función de Autocorrelación Simple.



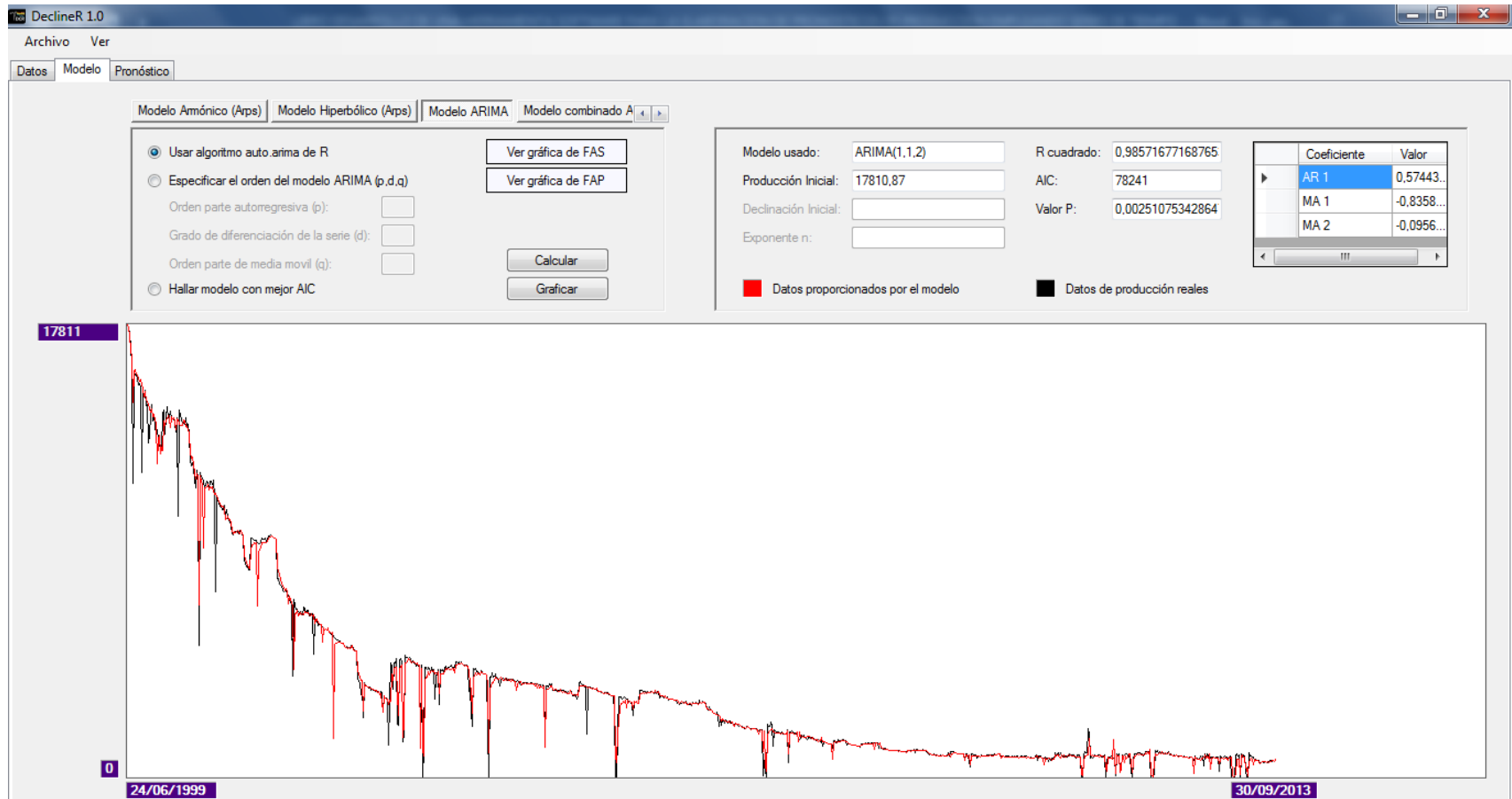
Fuente: R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Figura 3.4.9: Función de Autocorrelación Parcial.



Fuente: R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Figura 3.4.10: Modelo ARIMA realizado con función auto.arima().



Nota: En la parte derecha del panel de resultados se muestran los coeficientes del modelo. Graficados en negro están los datos reales, y en rojo el modelo.

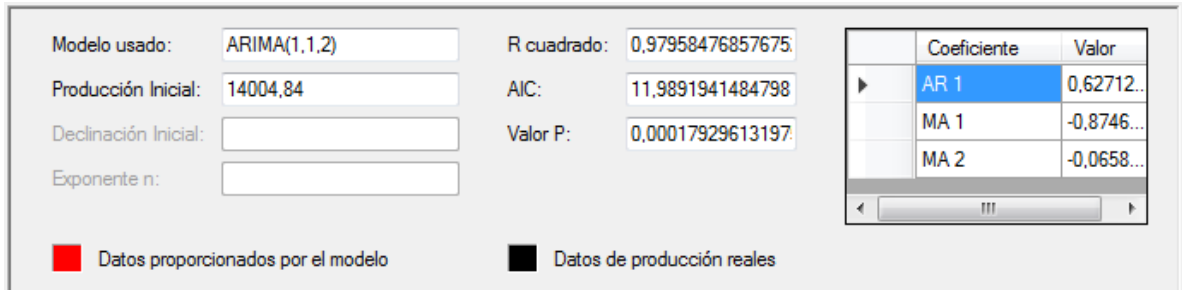
3.4.5. Modelo combinado: Realiza un modelo combinando el modelo hiperbólico automático con un modelo ARIMA dado por la función `auto.arima` de R.

3.4.6. Resultados del modelo: Los resultados del modelo construido se pueden ver en el panel de resultados (ver Figura 3.4.9). En la primera columna se presentan los parámetros del modelo según aplique, en la segunda se encuentran los criterios de selección:

- R cuadrado: Es el coeficiente de determinación, mide el ajuste del modelo. Debe buscarse un valor cercano a 1.
- AIC: El criterio de información de Akaike proporciona un valor que describe la pérdida de información, evaluando además del ajuste, la cantidad de parámetros del modelo. Su principal utilidad está en la elección del modelo ARIMA, ya que es en este donde la cantidad de parámetros varía según el orden. Debe buscarse el menor valor posible.
- Valor P (Sólo para ARIMA): El valor probabilístico evalúa la calidad del ajuste del modelo ARIMA mediante un análisis de los residuos obtenidos. Debe buscarse el mayor valor posible.

En la parte derecha del panel de resultados se puede observar una tabla que se utiliza solo con modelos ARIMA, para proporcionar el valor de los coeficientes del modelo (ver Figura 3.4.10).

Figura 3.4.11: Panel de resultados del modelo (Resultados de un modelo ARIMA).



Modelo usado: R cuadrado:

Producción Inicial: AIC:

Declinación Inicial: Valor P:

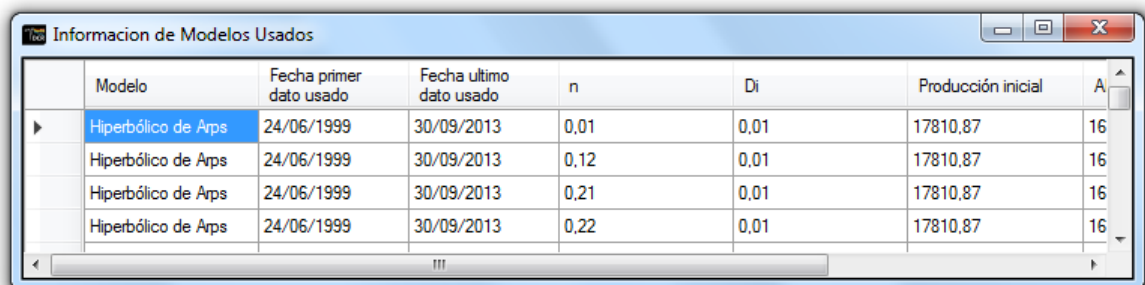
Exponente n:

	Coficiente	Valor
▶	AR 1	0,62712...
	MA 1	-0,8746...
	MA 2	-0,0658...

Datos proporcionados por el modelo Datos de producción reales

Los resultados de cada modelo van siendo almacenados para facilitar la selección final del modelo óptimo. En cualquier momento puede ver los resultados siguiendo la ruta ver/Ver información de modelos usados, enseguida se desplegará una tabla con la información de todos los modelos empleados en la sesión (ver Figura 3.4.11).

Figura 3.4.12: Información de los modelos usados.



	Modelo	Fecha primer dato usado	Fecha ultimo dato usado	n	Di	Producción inicial	A
▶	Hiperbólico de Aps	24/06/1999	30/09/2013	0,01	0,01	17810,87	16
	Hiperbólico de Aps	24/06/1999	30/09/2013	0,12	0,01	17810,87	16
	Hiperbólico de Aps	24/06/1999	30/09/2013	0,21	0,01	17810,87	16
	Hiperbólico de Aps	24/06/1999	30/09/2013	0,22	0,01	17810,87	16

La información contenida en esta tabla puede ser copiada (seleccionando todo y usando Ctrl + c) y pegada directamente en un archivo de Word, Excel, o cualquier otra herramienta de manejo de texto (Power point, Bloc de notas, etc).

La información de los modelos usados se perderá cuando se cierre el programa, o cuando se carguen datos nuevamente, ya que al cambiar los datos se considera que inicia una nueva sesión (para evitar confusión al tener modelos realizados para datos distintos).

3.5. PRONÓSTICO

Una vez definido el modelo con el que se va a trabajar (el último construido), pase a la página “Pronóstico” (ver Figura 3.5.4).

3.5.1. Generar pronóstico: En la parte superior izquierda están los controles para la generación del pronóstico y del informe. El número de intervalos de tiempo a pronosticar se debe ingresar en el cuadro de texto (el intervalo de tiempo es el que se eligió en la página datos, sección 3.3) y debe ser un número entero positivo.

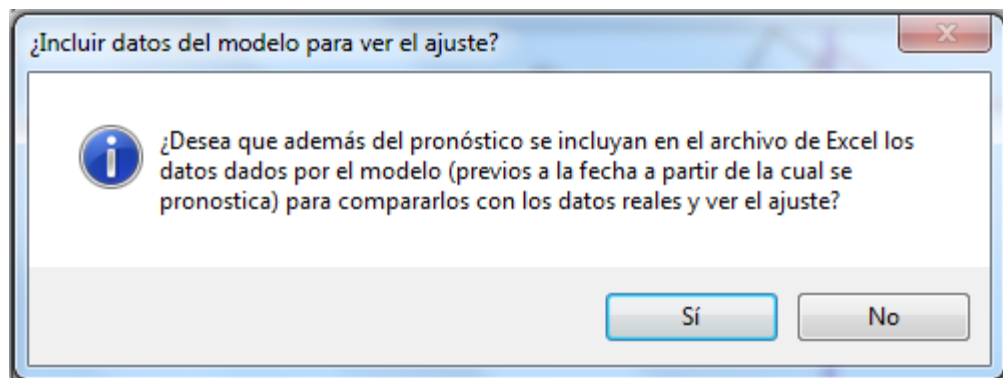
3.5.2. Generar informe: El informe es un archivo de Excel con los datos seleccionados (ver Figura 3.5.8)., la gráfica respectiva y los datos del modelo que se usó. Para la generación del informe se encuentran tres opciones principales:

- Generar informe con el pronóstico únicamente (ver Figura 3.5.5): Genera el archivo de Excel y le pasa sólo el pronóstico, por tanto, la gráfica y la tabla no contendrán ningún dato real de producción
- Generar informe incluyendo los datos que fueron usados (ver Figura 3.5.6): Genera el archivo de Excel con todos los datos que se usaron para construir el modelo (los datos seleccionados en la página de datos, sección 3.3.2) y los incluye en el gráfico.
El gráfico generado con ésta opción puede no dar una imagen significativa de los datos pronosticados (principalmente a tiempos cortos), ya que los datos usados (normalmente muchos más que los pronosticados, en proporción de aproximadamente cien a uno) no permite visualizar claramente los pronosticados.
- Generar informe incluyendo los últimos “j” datos usados (ver Figura 3.5.7): Genera el archivo de Excel con los últimos “j” (el valor de “j” debe ser un entero positivo y se especifica en el cuadro de texto indicado) datos de los que se usaron para construir el modelo (los datos seleccionados en la página de datos, sección 3.3.2) y los incluye en el gráfico.
Es la mejor opción para ver el ajuste del modelo y comparar visualmente la tendencia de los datos con la del pronóstico. Recuerde que si el valor de j es muy grande no se podrá ver adecuadamente el comportamiento del pronóstico.

Con las dos últimas opciones se generará en la ventana de DeclineR 1.0 el gráfico de los datos dados por el modelo (en rojo) superpuesto al de los datos reales (en negro), con el fin de facilitar el análisis visual del ajuste, una barra púrpura que indica el final de los datos reales y el inicio del pronóstico (La fecha correspondiente se muestra en el cuadro de texto “Fecha último dato”) en el panel de datos del modelo.

Para obtenerlo haga clic en el botón “Exportar a Excel”. Tenga en cuenta que el proceso puede durar algunos segundos. En el momento de generar el informe se le preguntará si desea incluir los datos generados por el modelo para ver el ajuste (ver Figura 3.5.1), esto es, los valores del modelo antes del último dato.

Figura 3.5.1: Opción para definir qué datos irán en el informe.



Si elige “Sí” los datos del modelo anteriores al pronóstico (todos en el caso de la segunda opción, o los últimos “j” en el caso de la tercera) serán tabulados en el archivo de Excel y obtendrá un gráfico como el de la figura 3.5.2, en caso contrario, para las fechas anteriores a la de inicio del pronóstico sólo se dará el dato real y el gráfico será como el de la figura 3.5.3.

Los gráficos de esta sección corresponden a un modelo ARIMA (4,1,3).

En el archivo de Excel las tablas de parámetros del modelo y de criterios de selección se encuentran ubicadas debajo del gráfico.

Figura 3.5.2: Gráfico de un modelo ARIMA que incluye los datos del modelo previos al pronóstico.

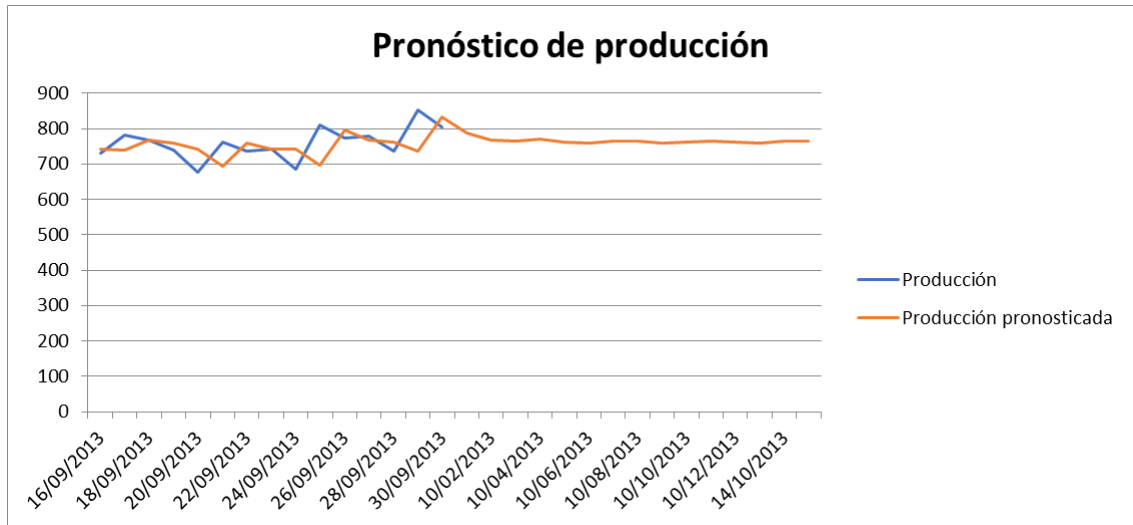


Figura 3.5.3: Gráfico de un modelo ARIMA que no incluye los datos del modelo previos al pronóstico.

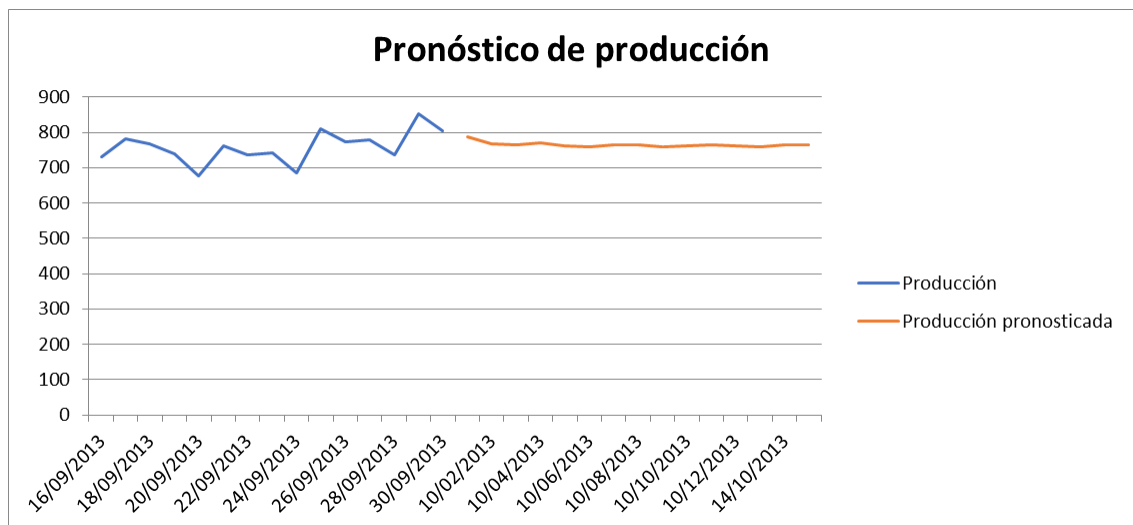
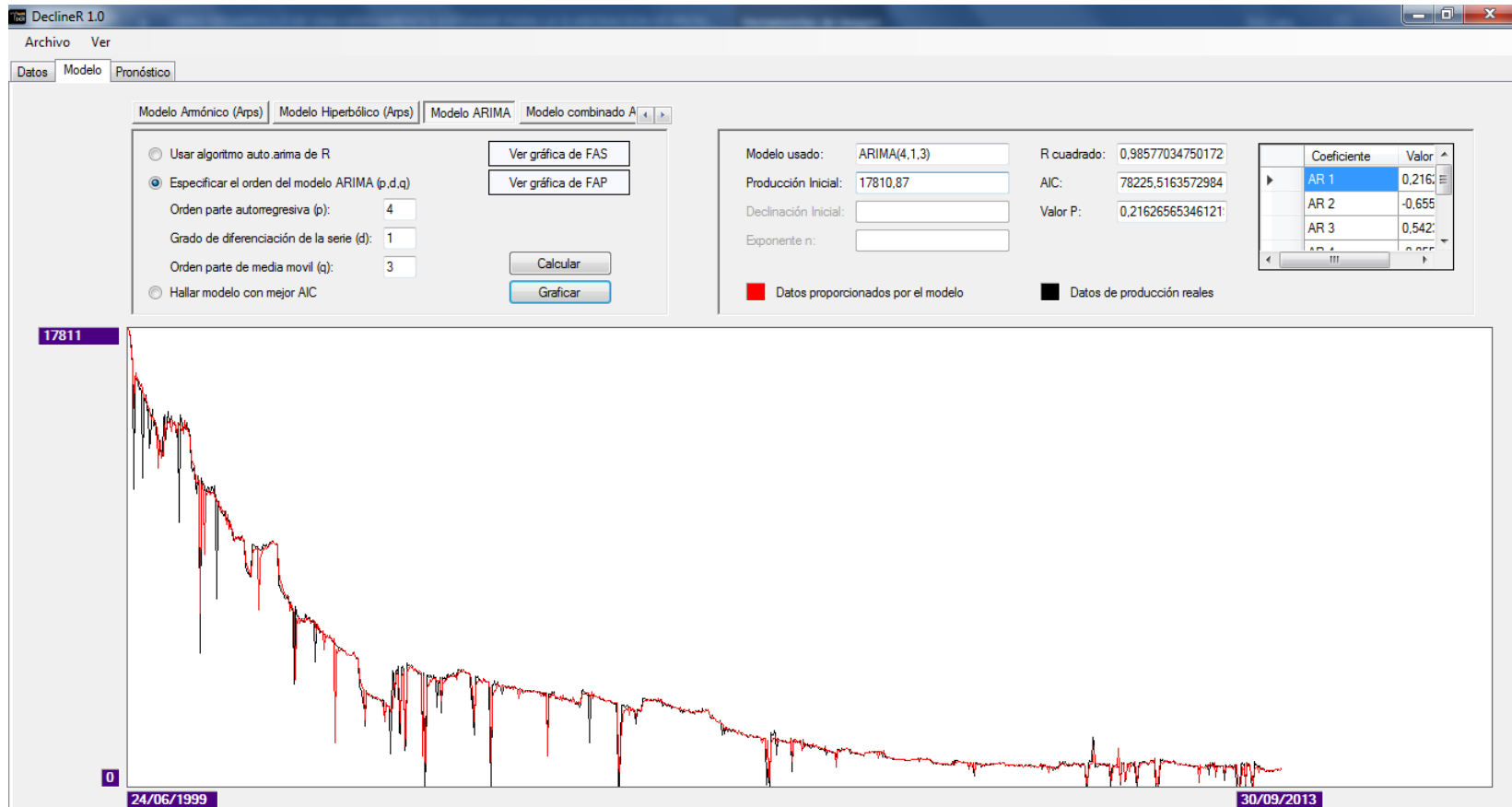


Figura 3.5.4: Página “Pronóstico”.



Nota: En la parte superior izquierda están los controles del Pronóstico y en la superior derecha los datos del modelo.

Figura 3.5.5: Generar informe con el pronóstico únicamente (quince días pronosticados).

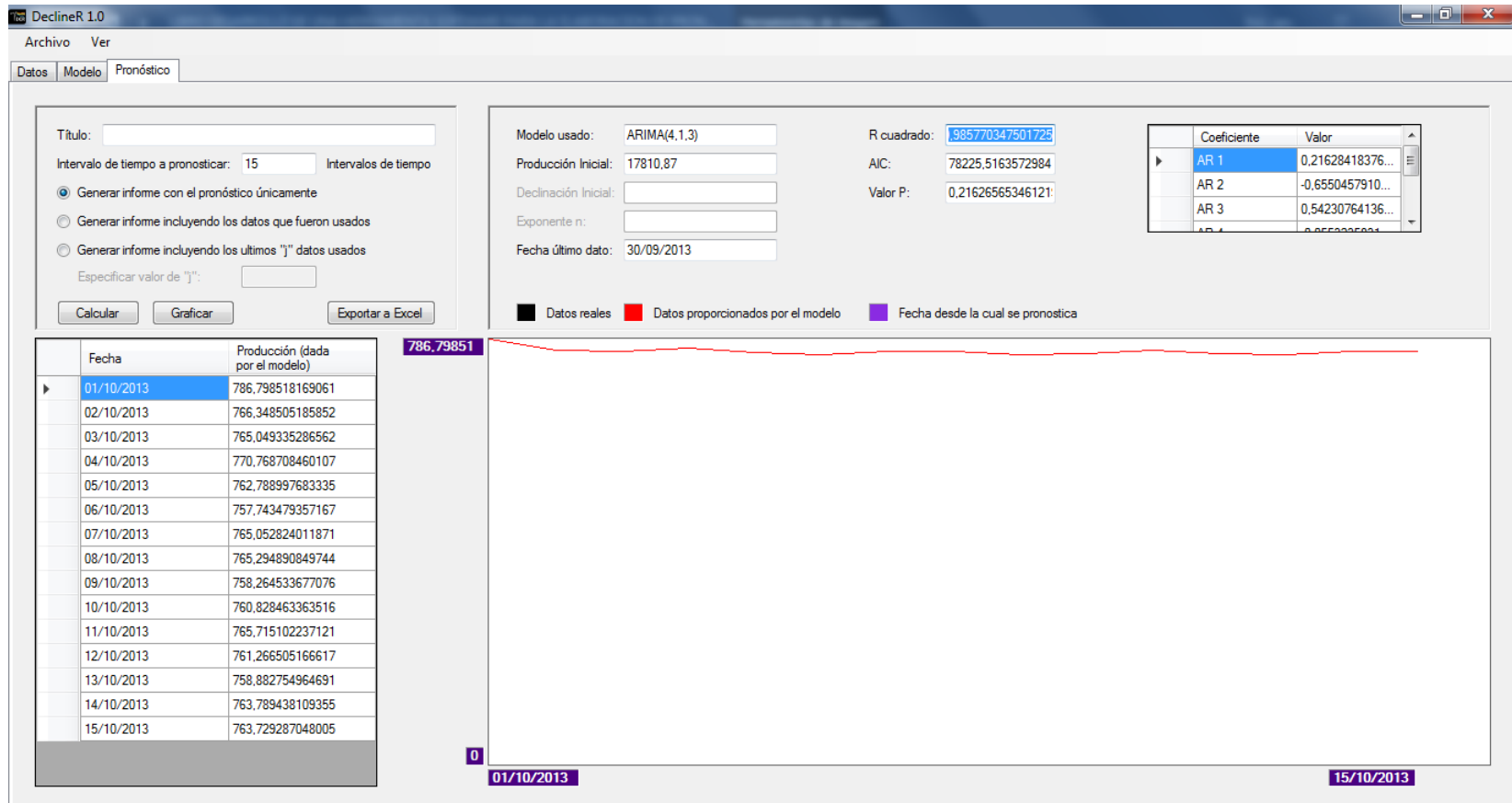
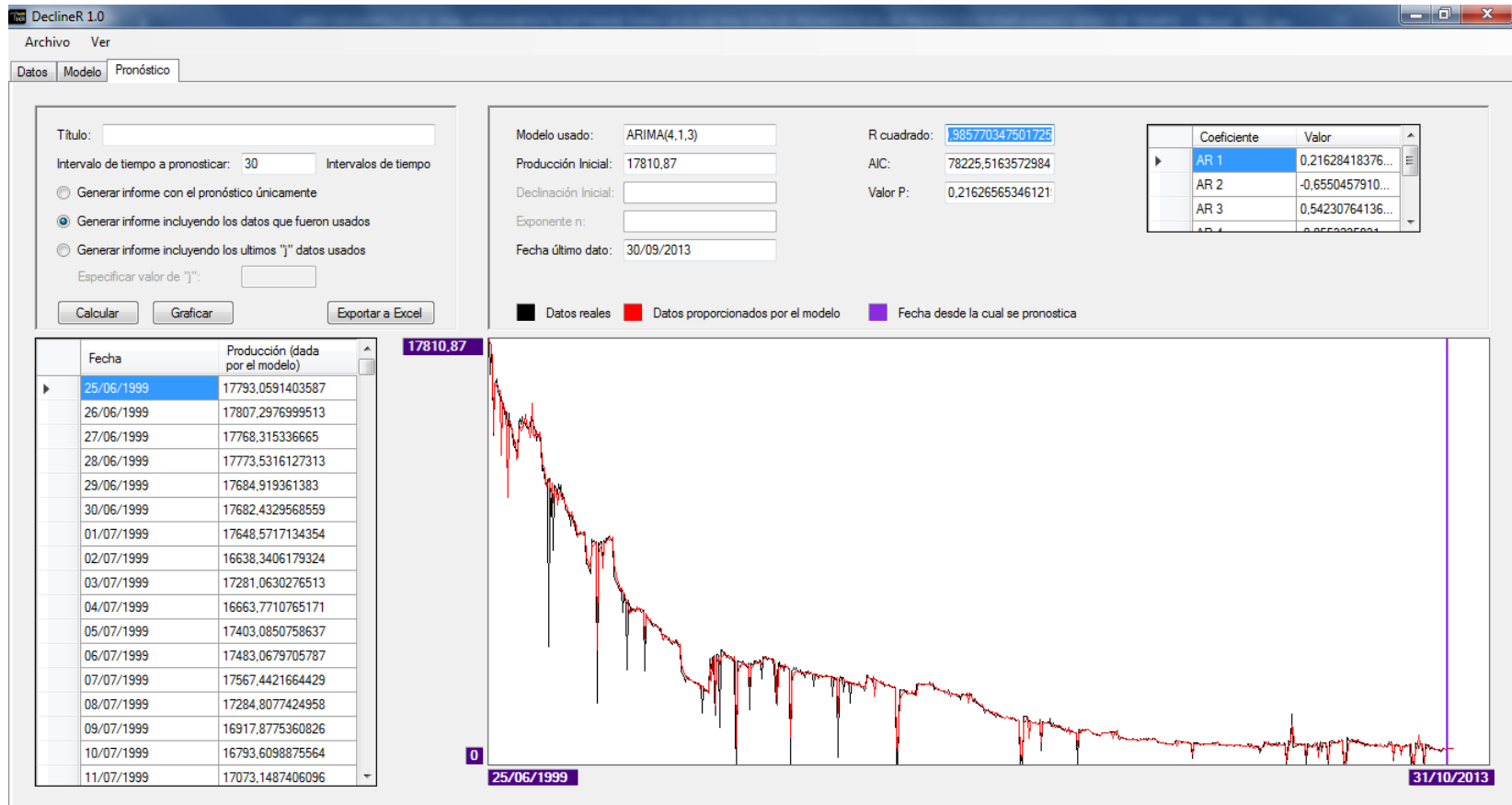


Figura 3.5.6: Generar informe incluyendo todos los datos que fueron usados (Treinta días pronosticados).



Nota: Pronóstico ubicado después de la barra púrpura. Nótese que no es posible ver claramente los datos pronosticados.

Figura 3.5.7: Generar informe incluyendo los últimos treinta datos que fueron usados (treinta días pronosticados).

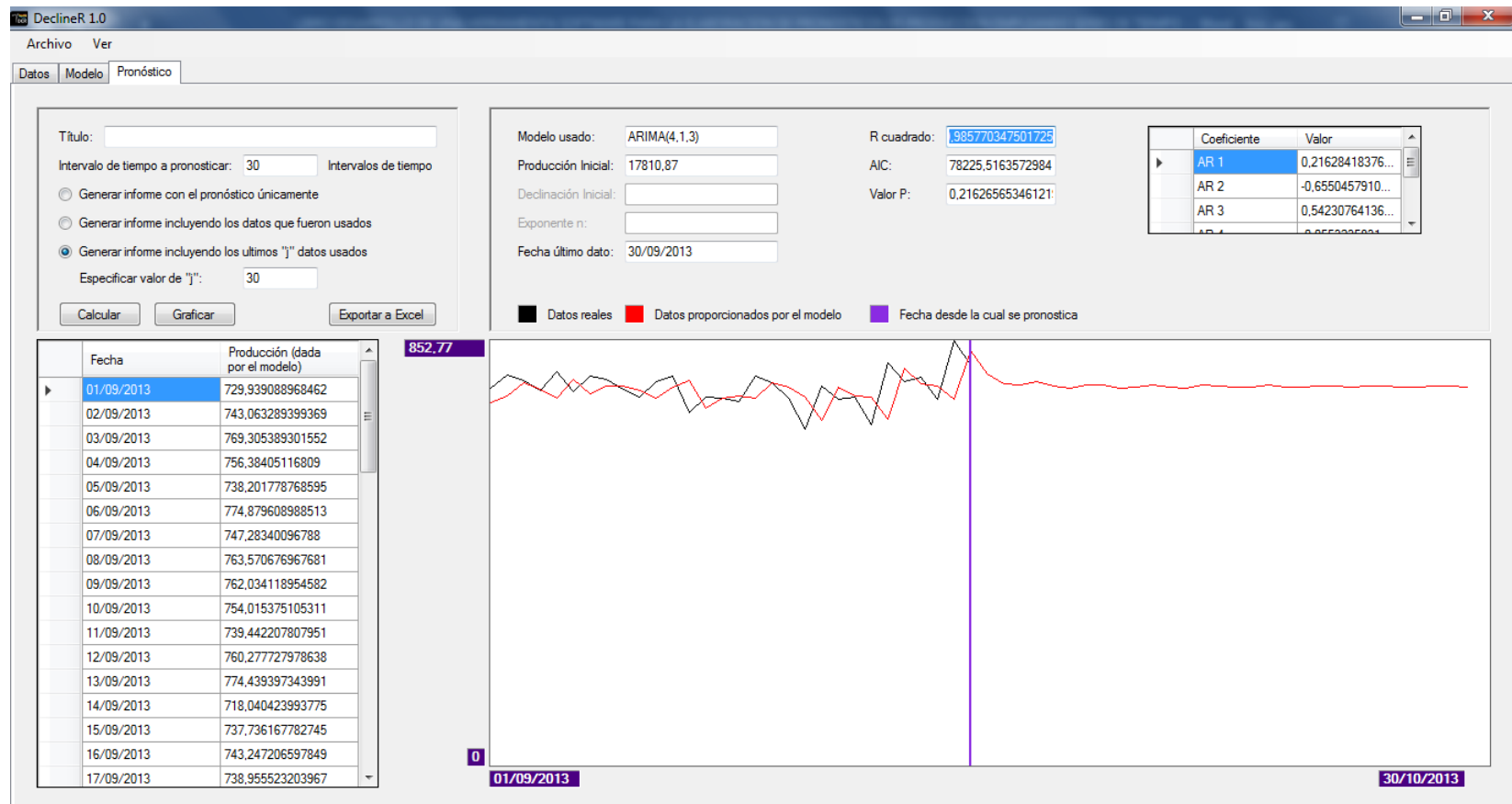
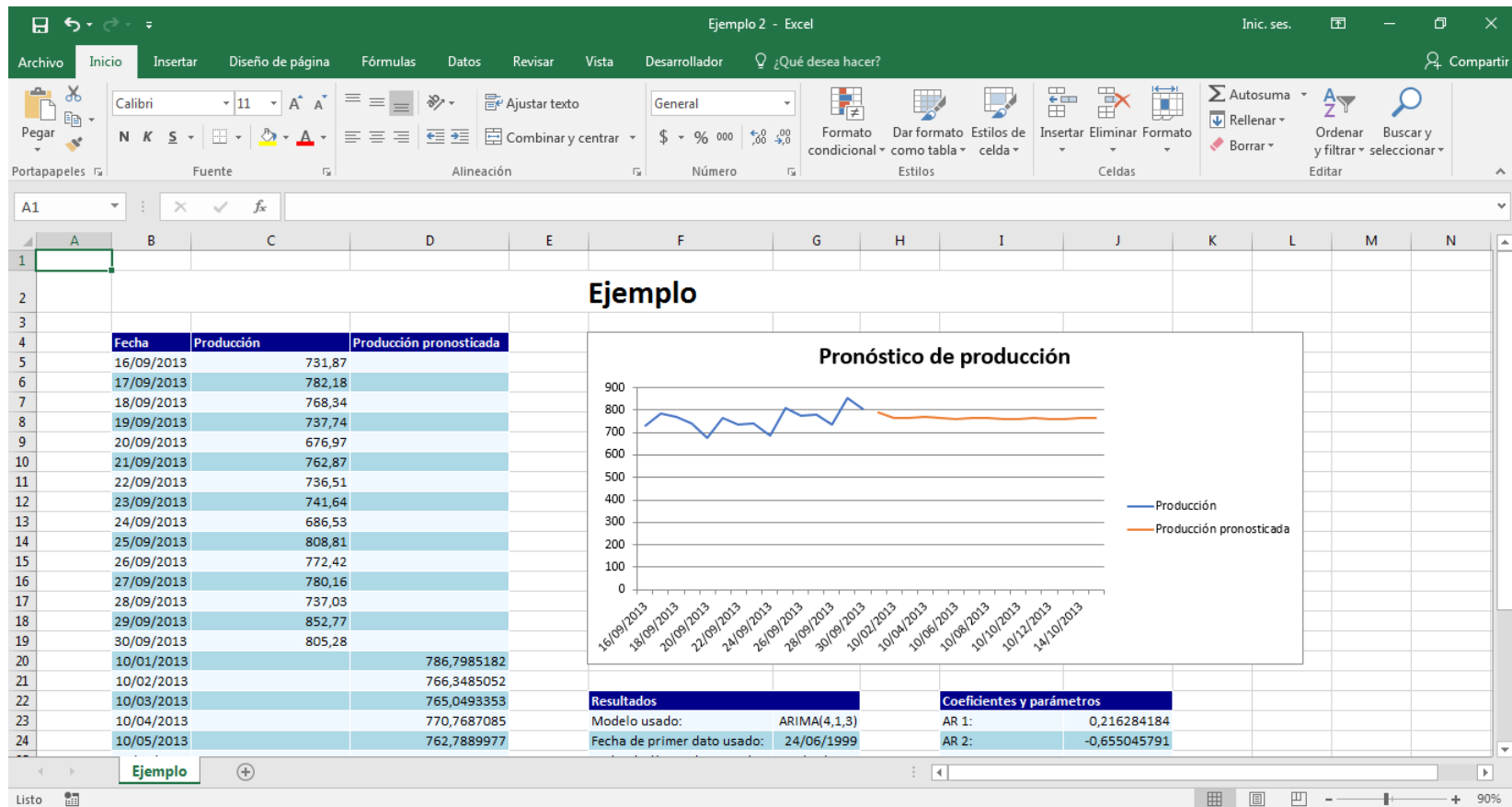


Figura 3.5.8: Ejemplo, informe generado pronosticando quince días e incluyendo los últimos quince datos utilizados.



Nota: No se incluyeron los datos dados por el modelo antes del pronóstico.

4. EJEMPLOS DE APLICACIÓN Y ANÁLISIS COMPARATIVO DEL ERROR

Según los resultados obtenidos por Montes Erik, la metodología de series de tiempo no es apta para la generación de pronósticos a mediano plazo (más de un año) o largo plazo, debido a que, después de los primeros 30 datos, ARIMA solo tenía en cuenta la componente tendencial.

Para el análisis se recopilaron datos de pozos utilizados en el trabajo de Antolinez B. y Carrillo R., y datos utilizados por Montes Erik, incluyendo los resultados de los pronósticos generados por las herramientas comerciales.

Los pozos utilizados son productores de aceite negro, con régimen de flujo pseudoestable (presentan pocos cierres de producción). Las gráficas de los historiales de producción se encuentran en el anexo A.

Debido a esto, el análisis que va a realizarse, se dividirá en dos secciones:

- Una sección de pronósticos a cortísimo plazo (un mes) para hacer un análisis comparativo del error obtenido por la metodología ARIMA y el modelo que combina ARIMA con el hiperbólico de Arps, incluida en el software y el obtenido por software comercial. Para este objetivo se utilizarán tres pozos con historia de producción de aproximadamente cinco años.
- Una sección de pronósticos utilizando los primeros seis, y luego los primeros diez años de vida productiva del pozo, para hacer un análisis comparativo entre error obtenido por el modelo hiperbólico automático y el modelo combinado incluidos en el software, y el obtenido por software comercial.

4.1. ANALISIS COMPARATIVO DEL ERROR A CORTÍSIMO PLAZO

4.1.1. Construcción de los modelos: Se procedió a cargar los datos al software DeclineR 1.0 y a construir modelos con cuatro métodos distintos, a continuación, se presentan los resultados.

Se debe aclarar que el AIC presentado en el modelo combinado corresponde únicamente a la parte ARIMA.

- Modelos para el pozo 1:

Tabla 4.1.1: Resultados de DeclineR para los modelos del pozo 1

Modelo	Fecha primer dato	Fecha último dato	n	Di	qi	AIC	R2
Combinado: Hiperbólico + ARIMA(1,1,1)	24/06/1999	30/09/2012	0,54961	0,0015446	17810,87	72693	0,98614
Hiperbólico de Arps	24/06/1999	30/09/2012	0,54961	0,0015446	17810,87		0,95794
ARIMA(1,1,1), auto.arima	24/06/1999	30/09/2012			17810,87	72722	0,98606
ARIMA(9,1,9), Mejor ARIMA con AIC	18/01/1999	30/09/2012			9239,55	77907	0,97694

- Modelos para el pozo 2:

Tabla 4.1.2: Resultados de DeclineR para los modelos del pozo 2

Modelo	Fecha primer dato	Fecha último dato	N	Di	qi	AIC	R2
Combinado: Hiperbólico + ARIMA(2,1,1)	08/05/1999	31/10/2012	0,56048	0,0025416	13512,16	69265	0,98387
Hiperbólico de Arps	08/05/1999	31/10/2012	0,56048	0,0025416	13512,16		0,89215
ARIMA(5,2,0), auto.arima	08/05/1999	31/10/2012			13512,16	71221	0,97596
ARIMA(8,1,8), Mejor ARIMA con AIC	08/05/1999	31/10/2012			13512,16	69239	0,98403

- Modelos para el pozo 3:

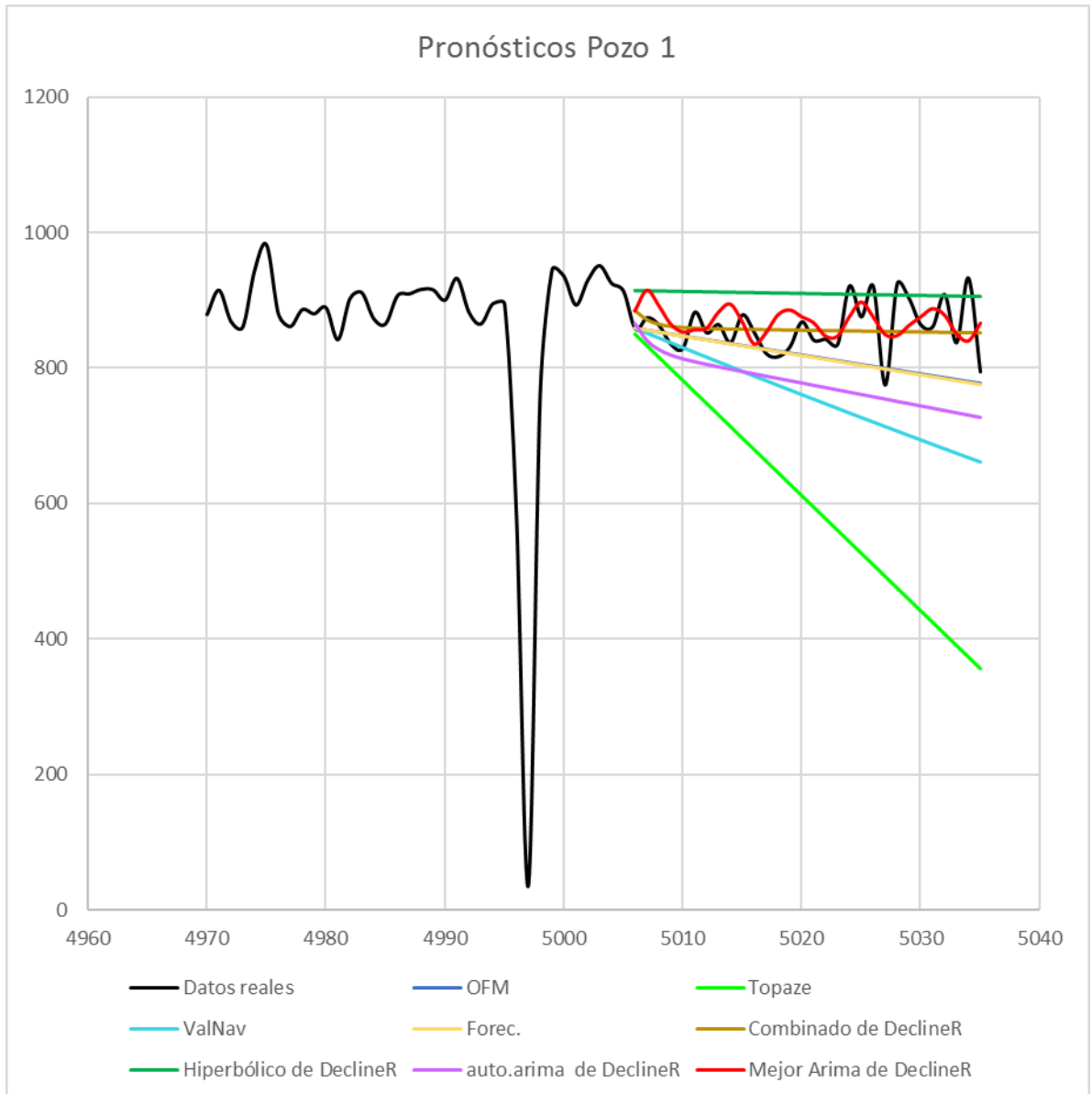
Tabla 4.1.3: Resultados de DeclineR para los modelos del pozo 3.

Modelo	Fecha primer dato	Fecha último dato	n	Di	qi	AIC	R2
Combinado: Hiperbólico + ARIMA(2,1,1)	25/06/1998	11/09/2012	0,96676	0,007196	20313,81	76367	0,97998
Hiperbólico de Arps	25/06/1998	11/09/2012	0,96676	0,007196	20313,81		0,87351
ARIMA(2,1,5), auto.arima	25/06/1998	11/09/2012			20313,81	76398	0,97989
ARIMA(8,1,8), -Mejor ARIMA con AIC	25/06/1998	11/09/2012			20313,81	76378	0,98003

4.1.2. Resultados obtenidos:

- Pozo 1:

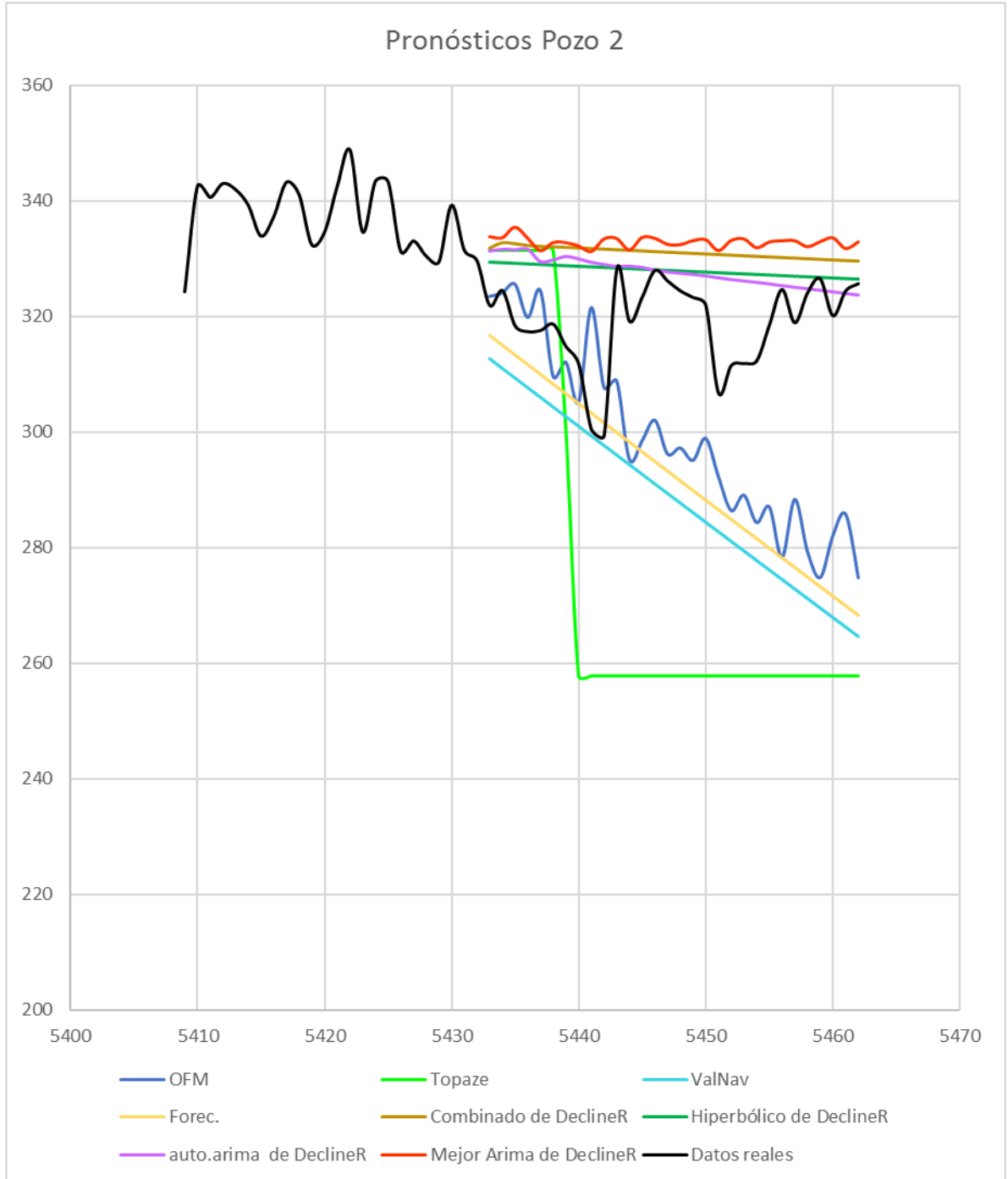
Figura 4.1.1: Grafico comparativo de pronósticos para el pozo 1.



Nota: Las gráficas de Forecast y OFM están superpuestas porque presentan los mismos resultados.

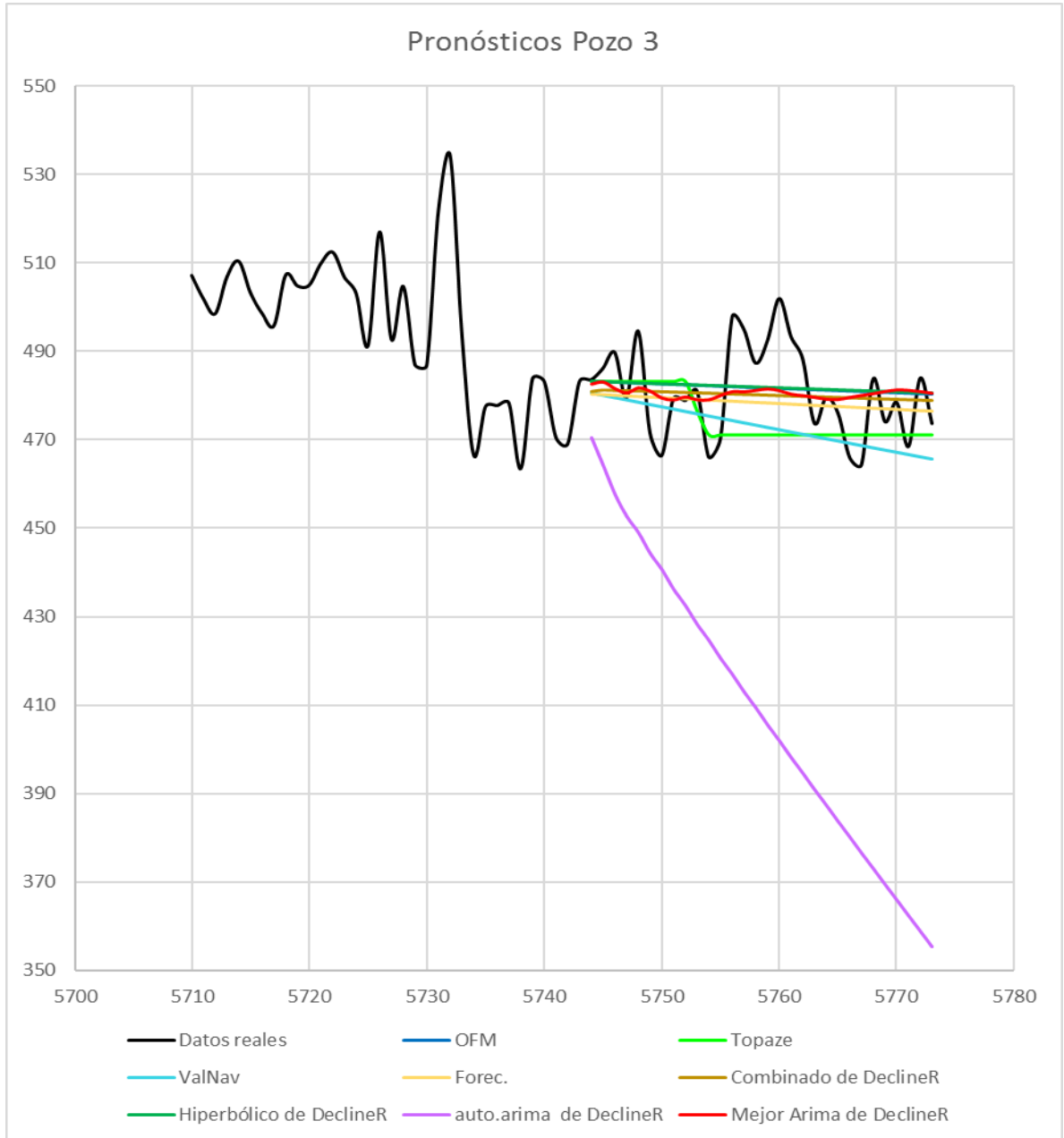
- Pozo 2:

Figura 4.1.2: Grafico comparativo de pronósticos para el pozo 2.



- Pozo 3:

Figura 4.1.3: Grafico comparativo de pronósticos para el pozo 3.



Nota: Las gráficas del modelo hiperbólico (de DeclineR 1.0) y OFM están superpuestas porque presentan los mismos resultados.

En el pozo 1 (Figura 4.1.1) puede observarse un muy buen ajuste del ARIMA con mejor AIC, y el modelo combinado (dados por DeclineR 1.0), destacando sobre los demás. El modelo hiperbólico también presentó un buen ajuste, mientras que el obtenido por auto.arima se aleja más de la realidad.

En el pozo 2 (Figura 4.1.2) los cuatro modelos dados por DeclineR 1.0 presentan buen ajuste, destacándose en este caso el modelo obtenido con auto.arima. El pronóstico dado por OFM tiene un buen ajuste al inicio, pero termina siguiendo una tendencia diferente que lo aleja de los datos reales.

En el pozo 3 (Figura 4.1.3) debe destacarse tres de los modelos dados por el software propuesto (todos menos el de auto.arima) en este trabajo de grado, el pronóstico dado por OFM, y el dado por Forecast tienen un alto nivel de ajuste. El modelo obtenido con auto.arima presenta, para este caso, una mala capacidad predictiva, tomando valores muy alejados de la realidad.

Puede observarse que los modelos hiperbólico, combinado y mejor ARIMA (modelo con mejor AIC) mantienen un buen ajuste para los tres pozos, destacándose los dos últimos en el pozo 1 y en el pozo 3.

El modelo hiperbólico, en general, presenta una buena capacidad predictiva.

El modelo obtenido con auto.arima obtuvo un buen ajuste en el pozo 2, pero en los pozos 1 y 3 presenta resultados distantes de la producción real.

En la tabla 4.1.4 se presenta el promedio de los errores absolutos y relativos presentados por los modelos para los tres pozos. Se debe destacar que el error relativo obtenido por el modelo combinado, y el obtenido como modelo ARIMA con mejor AIC, no supera el 5% en ninguno de los tres casos.

Tabla 4.1.4: Errores obtenidos por los modelos para todos los pozos.

	Pozo 1		Pozo 2		Pozo 3	
	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)
OFM	-42,6416667	5,343224125	-19,8283333	7,184809275	0,880666667	1,74690603
Topaze	-256,598	29,5828466	-44,8046667	15,50641038	-5,92	2,04684257
ValNav	-101,637	11,63854433	-30,2226667	9,410308497	-7,87966667	2,14197719
Forecast	-43,3426667	5,388210472	-26,339	8,305007764	-2,47433333	1,70673793
Hiperbólico, DeclineR 1.0	49,58156323	6,47195741	9,198800157	2,943617883	0,942038868	1,75164301
auto.arima, DeclineR 1.0	-81,3990265	9,323030303	8,881959446	2,933845899	-72,0535734	14,9774303
Combinado, DeclineR 1.0	-3,68622312	3,502481451	12,2163686	3,890886197	-0,78565317	1,70923006
Mejor ARIMA, DeclineR 1.0	8,382344905	3,965896961	14,03773089	4,459348293	-0,34288774	1,65322538

4.2. ANÁLISIS DEL ERROR OBTENIDO A MEDIANO PLAZO

Para este análisis se utilizó el pozo 1. Se realizaron dos pruebas para evaluar la capacidad predictiva de los modelos hiperbólico y combinado (planteados en el

software) a mediano plazo, y comparar el error obtenido con el de software comercial. Los dos casos son:

- Caso 1: Pronósticos a uno, dos, tres años y 8,6 años (final de los datos que se tienen para comparar) utilizando los primeros seis años del historial de producción.
- Caso 2: Pronósticos a uno, dos, tres años y 4,6 años (final de los datos que se tienen para comparar) utilizando los primeros diez años del historial de producción.

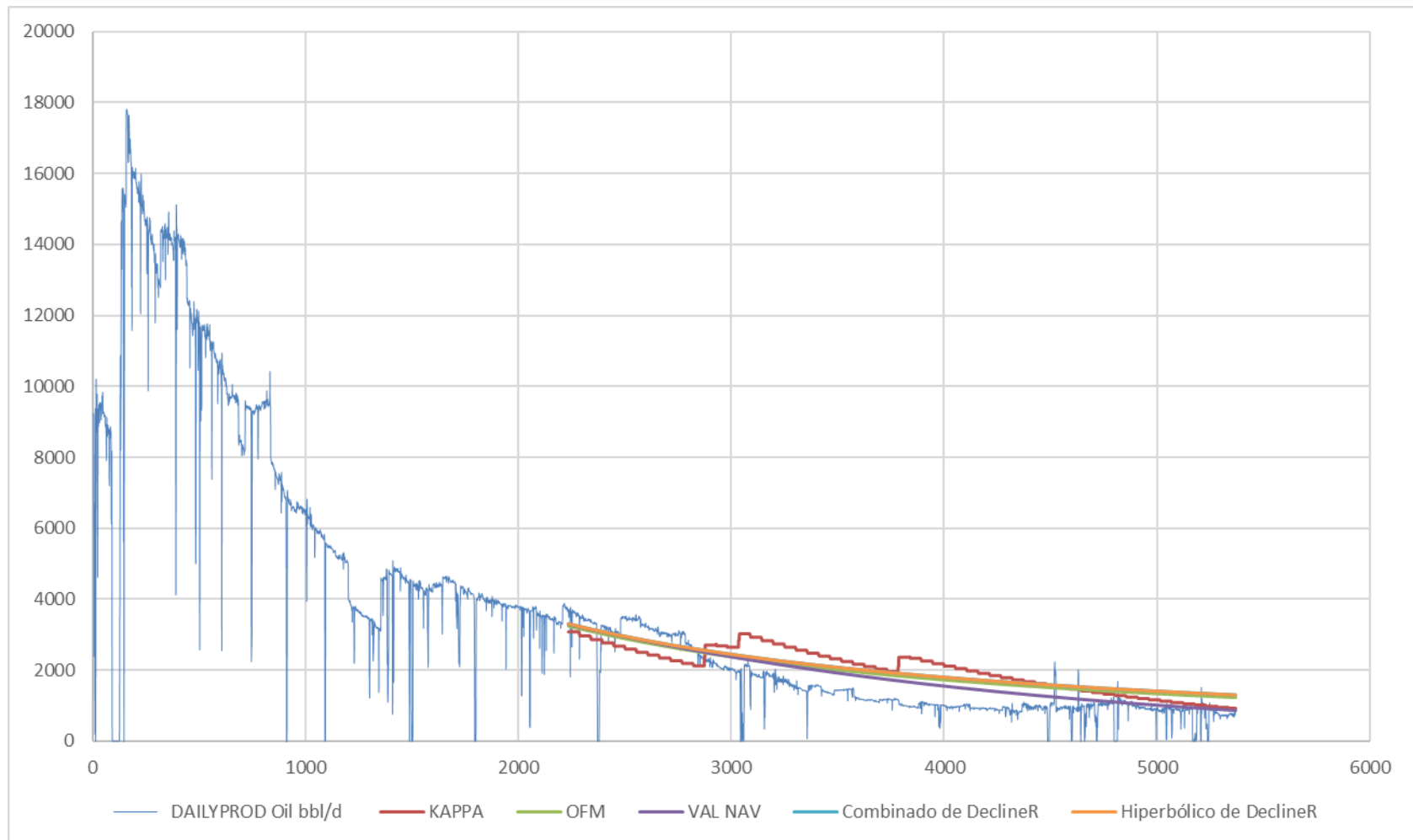
4.2.1. Caso 1:

4.2.1.1. Construcción de los modelos: Se procedió a cargar los datos al software DeclineR 1.0 y a construir modelos con dos métodos distintos, a continuación, se presentan los resultados (el AIC presentado en el modelo combinado corresponde únicamente a la parte ARIMA).

Tabla 4.2.1: Resultados de DeclineR para los modelos del Caso 1.

Modelo	Fecha primer dato	Fecha último dato	N	Di	qi	AIC	R2
Combinado: Hiperbólico + ARIMA(1,1,1)	24/06/1999	28/01/2005	0,81709	0,0017691	17810,87	32297	0,97263
Hiperbólico de Arps	24/06/1999	28/01/2005	0,81709	0,0017691	17810,87		0,92353

Figura 4.2.1: Pronósticos para el caso 1.



Nota: Las curvas de OFM y los dos modelos de DeclineR están superpuestas.

Tabla 4.2.2: Errores presentados por los pronósticos para el caso 1.

	Pronóstico a 1 año		Pronóstico a 2 años		Pronóstico a 3 años		Hasta el final de los datos (8,6 años)	
	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)
KAPPA	-520,165742	19,546265	-426,506247	20,584782	52,7518022	43,971657	487,486706	101,725249
OFM	-302,316515	13,8591254	-216,022155	12,2657364	20,6161792	26,0293795	402,252268	91,8556698
VAL NAV	-249,709074	12,5557389	-175,335998	11,338189	39,9013658	24,9030819	262,267918	67,3090335
DeclineR Combinado	-234,110618	12,160692	-145,650298	11,2808977	92,6316062	27,0726826	479,063775	99,7090541
DeclineR Hiperbólico	-238,952726	12,2804414	-150,498332	11,3448828	87,781644	27,0057377	474,114803	99,2143589

4.2.1.2. Resultados obtenidos: Los resultados para este caso se muestran en la figura 4.2.1 y la tabla 4.2.2

Observaciones:

- El mejor ajuste fue el de ValNav.
- Los modelos hiperbólico y combinado de DeclineR muestran pronósticos similares al obtenido con OFM, pero evaluando el error obtenido (que también es similar) se destaca el modelo ARIMA combinado para los primeros dos años, sin embargo, a partir del tercer año baja la calidad del pronóstico siendo superado por OFM.

4.2.2. Caso 2:

4.2.2.1. Construcción de los modelos: Se procedió a cargar los datos al software DeclineR 1.0 y a construir modelos con dos métodos distintos, a continuación, se presentan los resultados (el AIC presentado en el modelo combinado corresponde únicamente a la parte ARIMA).

Tabla 4.2.2: Resultados de DeclineR para los modelos del Caso 2.

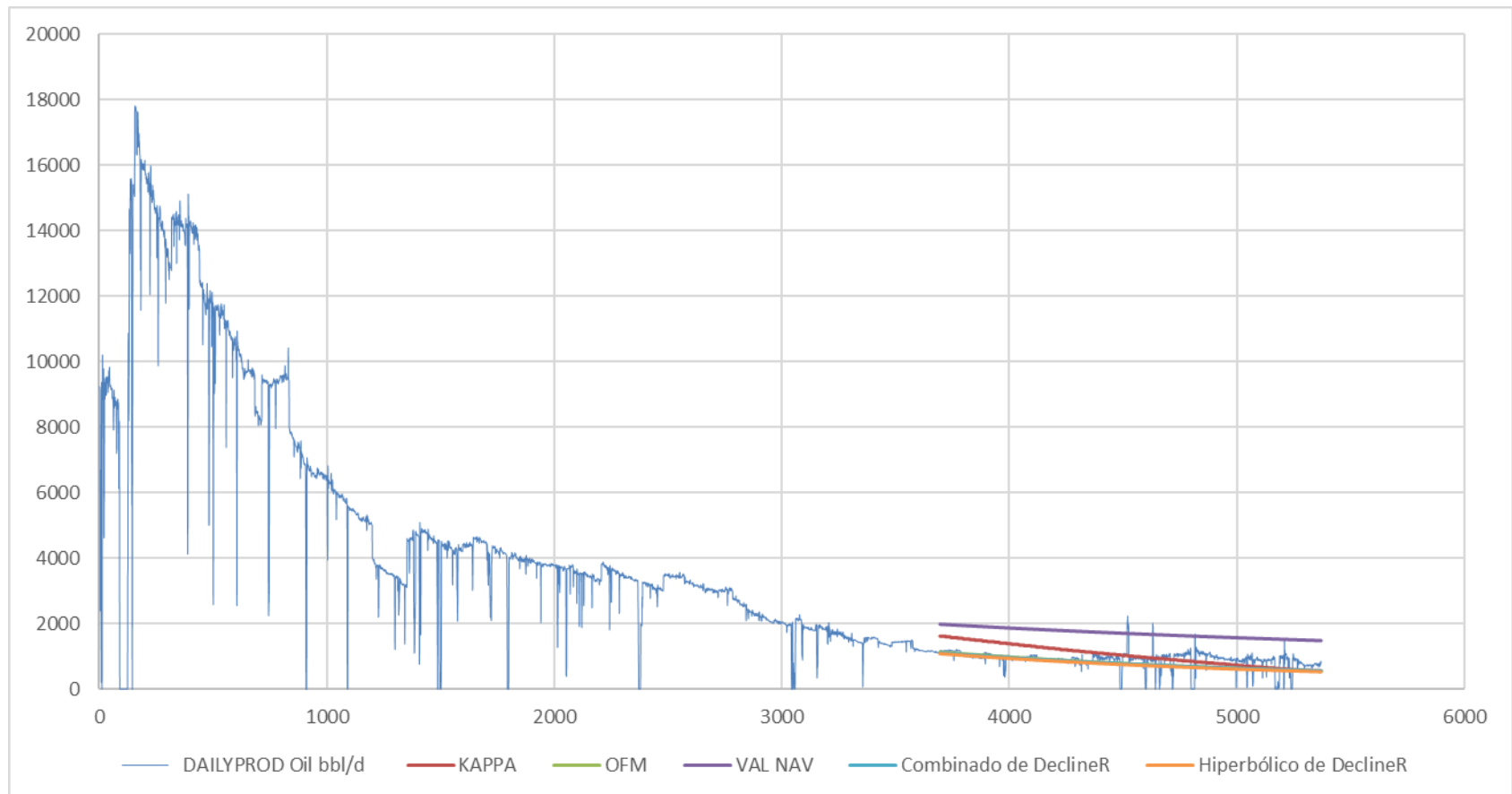
Modelo	Fecha primer dato	Fecha último dato	N	Di	qi	AIC	R2
Combinado: Hiperbólico + ARIMA(1,1,1)	24/06/1999	27/02/2009	0,37497	0,0014014	17810,87	54073	0,98232
Hiperbólico de Arps	24/06/1999	27/02/2009	0,37497	0,0014014	17810,87		0,9366

4.2.2.2. Resultados obtenidos: Los resultados para este caso se muestran en la figura 4.2.2 y la tabla 4.2.3.

Observaciones:

- Los modelos hiperbólico y combinado de DeclineR muestran pronósticos similares al obtenido con OFM, siendo los tres pronósticos con mejor ajuste, pero evaluando el error obtenido (que también es similar), OFM tiene el mejor ajuste.
- Se destaca entre los modelos propuestos en DeclineR el ajuste dado por el modelo combinado, que resulta tener un error relativo muy similar al de OFM (menos de 1% de diferencia en todos los casos).

Figura 4.2.2: Pronósticos para el caso 2.



Nota: Las curvas de OFM y los dos modelos de DeclineR están superpuestas.

Tabla 4.2.2: Errores presentados por los pronósticos para el caso 1.

	Pronóstico a 1 año		Pronóstico a 2 años		Pronóstico a 3 años		Hasta el final de los datos (4,6 años)	
	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)	Error absoluto	Error relativo (%)
KAPPA	451,983562	46,4282842	366,953452	38,8406735	249,414766	79,4176264	107,230388	76,0682585
OFM	12,1605687	7,49738473	-16,065754	6,90722386	-79,0962806	50,3833529	-142,943441	57,1294715
VAL NAV	883,306156	89,6226724	872,465821	92,1316319	821,530924	174,319809	796,267462	177,344678
DeclineR Combinado	-8,10293362	7,78345131	-32,2808028	7,51846451	-91,3757819	50,7571252	-149,533225	57,3168916
DeclineR Hiperbólico	-34,9955723	8,51777418	-59,2065293	9,09547867	-118,312273	51,1158283	-177,277697	57,8844907

5. CONCLUSIONES

En los pozos estudiados, la metodología planteada para encontrar el mejor modelo ARIMA con base en el criterio de información de Akaike resultó ser muy útil en la elaboración de pronósticos de producción a cortísimo plazo, obteniendo ajustes que superaron a las herramientas comerciales utilizadas.

Se hizo evidente con los resultados que los modelos generados utilizando la función `auto.arima()` del software R fueron superados en la mayoría de las ocasiones por los encontrados con el algoritmo diseñado para automatizar la metodología presentada por Montes E.⁶⁴ que busca el mejor modelo ARIMA.

Al tomar la serie de tiempo como estacionaria en tendencia y combinar el modelo hiperbólico (para describir la componente tendencial) con el ARIMA (para describir las componentes cíclicas y estacional) hallado por la función `auto.arima()` de R se obtuvo un buen ajuste tanto en los pronósticos a cortísimo plazo (un mes), como en los realizados a uno y dos años.

La curva hiperbólica de Arps resultó ser la mejor opción presentada por el software para realizar pronósticos a tres o más años en los casos analizados. Cabe destacar que el ajuste obtenido fue muy similar al que presenta el software OFM.

Para pozos que lleven poco tiempo desde el inicio de la declinación (menos de dos años), la mejor opción planteada es la de modelo hiperbólico semiautomático siguiendo los valores del exponente de declinación planteado por Fetkovich⁶⁵ para determinados tipos de yacimientos.

⁶⁴ MONTES, Erick. Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos. Trabajo de grado (Magíster en Ingeniería de Hidrocarburos) Universidad Industrial de Santander. Bucaramanga. 2016.

⁶⁵ FETKOVICH, M.J., FETKOVICH, E.J., & FETKOVICH, M.D. Useful Concepts for Decline-Curve Forecasting, Reserve Estimation, and Analysis. Paper SPE 28628. 1994

No se recomienda usar modelos que involucren las curvas empíricas de Arps en pozos cuya declinación aun no haya iniciado. En estos casos la mejor opción a usar corresponde al modelo ARIMA, teniendo en cuenta que los resultados solo serán significativos a cortísimo plazo, y que este modelo no predecirá el inicio de la declinación.

6. RECOMENDACIONES

Se recomienda incursionar en el amplio número de paquetes y librerías que tiene disponibles el lenguaje de programación R y aplicarlo en distintas áreas de la ingeniería de petróleos.

Se recomienda ampliar el alcance de este software, adicionándole nuevas opciones para construcción de modelos, tales como vectores autorregresivos en el caso de la estadística, o redes neuronales, y mejorando los métodos planteados en el presente trabajo.

Se recomienda estudiar la posibilidad de realizar el proceso de identificación del modelo ARIMA con mejor AIC en varios hilos, para aprovechar al máximo los recursos disponibles en el ordenador en que se esté trabajando, disminuyendo así el tiempo de ejecución y logrando resultados en menor tiempo.

BIBLIOGRAFIA

ABDELHAFIDH, F. y DJEBBAR, T.: "Application of decline-curve analysis technique in oil reservoir using a universal fitting equation", paper SPE 70036, (May 2001).

AGARWAL, R.G., et al. Analyzing well production data using combined type-curve and decline-curve analysis concepts, paper SPE 49222, 1998.

ANTOLÍNEZ, Brayan. CARRILLO, Reynel. ESTUDIO COMPARATIVO DE LOS MODELOS UTILIZADOS EN LA GENERACIÓN DE PRONÓSTICOS DE PRODUCCIÓN PARA YACIMIENTOS CON FLUIDOS COMPOSICIONALES. Trabajo de grado. Bucaramanga: Universidad Industrial de Santander, 2015.

ANDERSON, D.R.; SWEENEY, D. J. y WILLIAMS T. A. Estadística para Administración y Economía. Mexico: Caengage Learning 10a. edición, 2008.

ARPS, J. Analysis of decline curves. The British-American Oil Producing Co. SPE 945228-G. 228-247.(1945)

ARNOLD, R. ANDERSON, R. Preliminary Report on Coalinga Oil District. U.S. Geological Survey Bulletin, 357 (1908) 79.

B. D., Ripley (2002). Time series in R 1.5.0. *R News*, 2/2, 2–7. Disponible en: https://www.r-project.org/doc/Rnews/Rnews_2002-2.pdf.

BLASINGAME, T.A. y LEE, W.J. Properties of Homogeneous Reservoirs, Naturally Fractured Reservoirs, and Hydraulically Fractured Reservoirs from Decline Curve Analysis, paper SPE 15018 presented at the 1986 SPE Permian basin Oil & Gas Recovery Conference, Midland, March 13-14.

BLASINGAME, T.A.; McCRAY, T.C. y LEE, W.J. Decline Curve Analysis for Variable Pressure Drop/Variable Flowrate Systems. Paper SPE 21513 presented at the 1991 SPE Gas Technology Symposium, Houston.

BOWERMAN, Bruce. O'CONNELL, Richard. KOEHLER, Anne. Pronósticos, series de tiempo y regresión, un enfoque aplicado. CENGAGE Learning. Cuarta edición. 2006.

BRITO, L. E.; PAZ F. y BELISARIO, D. R. "Probabilistic Production Forecasts Using Decline Envelopes", paper SPE 152392-MS, (2012).

CUTLER, W.W. Jr. Estimation of Underground Oil Reserves by Well Production Curves. U.S. Bureau of Mines Bulletin, 228 (1924).

DUONG, A.N. A New Approach for Decline-Curve Analysis. Paper SPE 18859 presented at the 1989 SPE Production Operations Symposium, Oklahoma, March 13-14.

ENERGY NAVIGATOR. Value Navigator®, Energy Navigator 2016. ALL RIGHTS RESERVED.

FETKOVICH, M.J. Decline Curve Analysis Using Type Curves. JPT (June, 1980) 1065-1077.

G.P.E. Box y G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden Day, San Francisco, 1978.

GENTRY, R.W. Decline Curve Analysis. JPT (January, 1972) 38-41.

GENTRY, R.W. y McCRAY, A.W. The Effect of Reservoir and Fluid Properties on Production Decline Curves. JPT (September, 1978) 1327-1341.

GUJARATI, Damodar. PORTER, Dawn. Econometría. Mc Graw Hill, Quinta edición. 2009.

HARRINGTON, A.G.; LEE, B.Y. y TAYLOR, P.S. Application of Type-Curve Techniques to Decline Analysis and Forecasting of Gas Wells, paper SPE 16936 presented at the 1987 Annual Technical Conference and Exhibition of the SPE, Dallas, TX, September 27-30

KAPPA. Petroleum Exploration & Production Software Training & Consulting. Topaze NL, V4.2. Kappa: 1987-2016.

MASONER, L.O. "Decline analysis relationship to relative permeability in secondary and tertiary recovery", paper SPE 39928, (April 1998).

McCRAY, T.C. Reservoir Analysis Using Production Decline Data and Adjusted Time. M.S. Thesis, Texas A&M U., College Station, TX (1990).

MONTES, Erick. Modelado de los Pronósticos de Producción a partir del Análisis Integrado de Datos y Métodos Estadísticos No Paramétricos. Trabajo de grado (Magíster en Ingeniería de Hidrocarburos) Universidad Industrial de Santander. Bucaramanga. 2016.

NOVALES, Alfonso. Econometría, 2 ed. Madrid: McGraw Hill, 1993.

PARIS DE FERRER, M. (2009). Fundamentos de Ingeniería de Yacimientos. Maracaibo: Ediciones Astro Data S.A. 2009.

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

RYDER SCOTT COMPANY, L.P. Forecast Multi-Trend Decline Analysis. Ryder Scott Company, L.P: 2016.

SCHLUMBERGER LIMITED. Oil Field Manager Well and Reservoir Analysis Software, 2009. Schlumberger 2016. ALL RIGHTS RESERVED.

SLIDER, H.C. A Simplified Method of Hyperbolic Decline Curve Analysis. JPT (March, 1968) 235-236.

ANEXOS

Anexo A. Historiales de producción de los pozos utilizados.

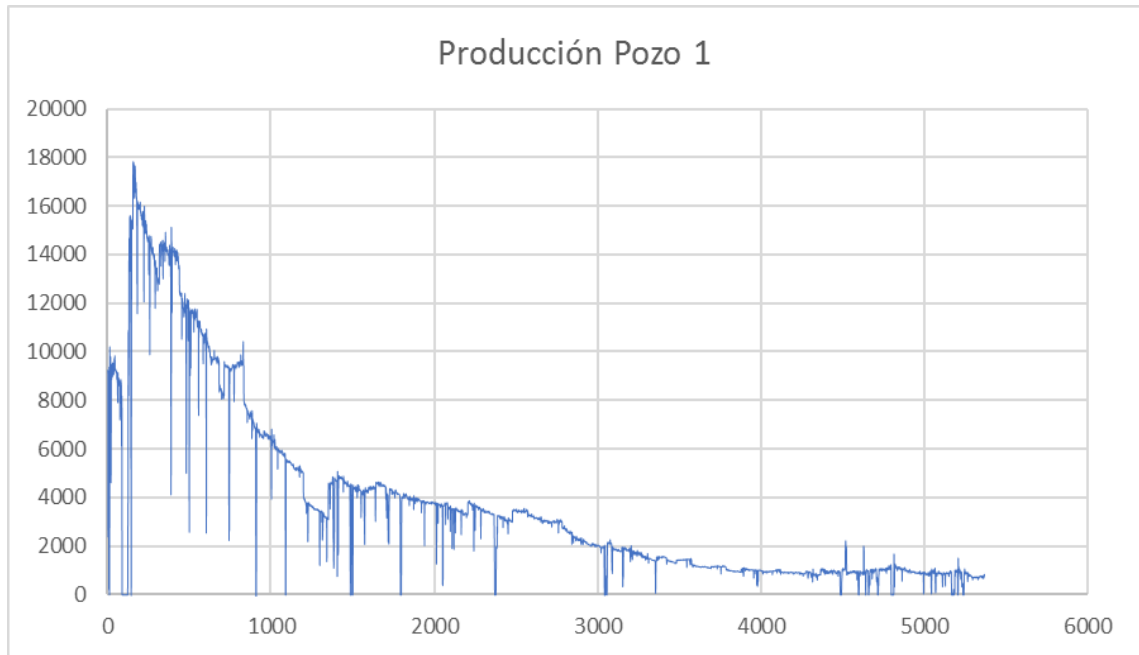


Figura A.1: Historial de producción del pozo 1.

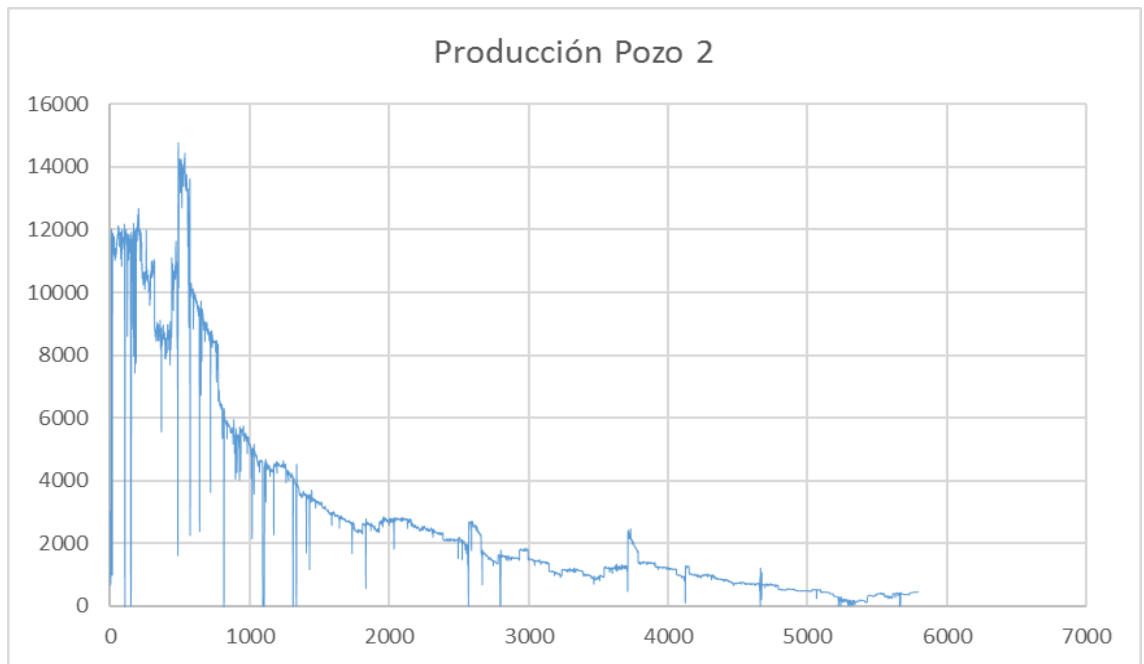


Figura A.2: Historial de producción del pozo 2.

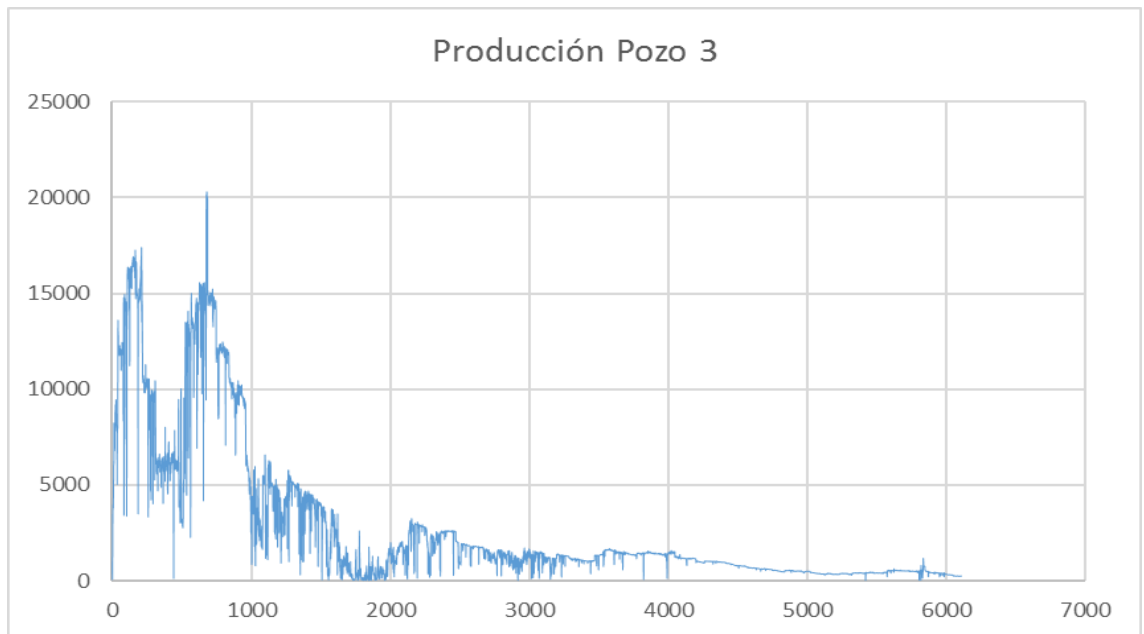


Figura A.3: Historial de producción del pozo 2.

Anexo B. Código de la case “Constructor de modelos”.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
using RDotNet;
using System.Windows.Forms;
using System.Threading;
using System.IO;
using System.Globalization;
```

```
namespace DeclineR.Pantalla_Principal
{
    class ConstructorDeModelos
    {
        protected double nMinimoParaNoSerCero = 0.005;

        public void ConstruirModeloHiperbolicoDandoDiYN(double[] vectorDeDatos, out
double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, double
n, double declinacionInicial, out double r2, out double aic)
        {
            int numeroDeParametros = 2;
            double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], ti = 0, tf = usarDatosHasta;
            double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
            if (Math.Abs(n) > nMinimoParaNoSerCero)
            {

                DarValoresAlModeloHiperbolico(ref                vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
            }
        }
    }
}
```

```

        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
    }
    else
    {
        DarValoresAlModeloExponencial(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
    }

    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}

public void ConstruirModeloHiperbolicoQuePasePorElUltimoPunto(double[]
vectorDeDatos, out double[] vectorValoresDelModelo, int usarDatosDesde, int
usarDatosHasta, double n,out double declinacionInicial,out double r2,out double aic)
{
    int numeroDeParametros = 2;
    double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], ti = usarDatosDesde, tf = usarDatosHasta -
usarDatosDesde;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

    if(Math.Abs(n) > nMinimoParaNoSerCero)
    {
        declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) / (n
* tf);

        DarValoresAlModeloHiperbolico(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
    }
    else

```

```

    {
        declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / tf;

        DarValoresAlModeloExponencial(ref          vectorLocalValoresDelModelo,
        produccionInicial, declinacionInicial);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
        usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
        usarDatosHasta, out aic, numeroDeParametros);
    }

    vectorValoresDelModelo = vectorLocalValoresDelModelo;

}

public void ConstruirModeloHiperbolicoAutomatico(double[] vectorDeDatos, out
double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, int
numeroDeIteraciones, out double declinacionInicial, out double n, out double r2, out
double aic)
{
    /*Se plantea que el modelo esté obligado a pasar por el punto inicial y final de los
datos
*/

    double a = 0.1, b = 1, produccionInicial = vectorDeDatos[usarDatosDesde],
    produccionFinal = vectorDeDatos[usarDatosHasta], ti=usarDatosDesde,
    tf=usarDatosHasta - usarDatosDesde;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
    double r2Anterior, paso = 0.1, na, nb, r2na, r2nb;
    int numeroDeParametros = 2;

    n = -1;
    if (Math.Abs(n) < nMinimoParaNoSerCero)
    {
        declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / (tf);
        n = 0;
    }
    else
    {

```

```

        declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) / (n
* tf);
    }

    DarValoresAlModeloHiperbolico(ref                vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
    HallarR2(vectorDeDatos,    vectorLocalValoresDelModelo,    usarDatosDesde,
usarDatosHasta, out r2);
    r2Anterior = r2;

    while (Math.Abs(r2) >= r2Anterior)
    {
        if(r2 >= 0)
        {
            r2Anterior = r2;
        }
        else
        {
            r2Anterior = 0;
        }

        n = n + paso;
        if (Math.Abs(n) < nMinimoParaNoSerCero)
        {
            declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / (tf);
            n = 0;
        }
        else
        {
            declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) /
(n * tf);
        }
        DarValoresAlModeloHiperbolico(ref                vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
        HallarR2(vectorDeDatos,    vectorLocalValoresDelModelo,    usarDatosDesde,
usarDatosHasta, out r2);
    }

    a = n - 2 * paso;
    b = n + paso;

    for (int iteracion = 0; iteracion <= numeroDeIteraciones; iteracion++)

```

```

{
  na = a + (b - a) / 4;
  nb = a + 3 * (b - a) / 4;
  if (na == 0)
  {
    declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / (tf);
  }
  else
  {
    declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal,
na)) / (na * tf);
  }

  DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, na);
  HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2na);

  if (nb == 0)
  {
    declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / (tf);
  }
  else
  {
    declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal,
nb)) / (nb * tf);
  }

  DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, nb);
  HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2nb);

  if (r2na > r2nb)
  {
    b = nb;
  }
  else
  {
    a = na;
  }
}

```

```

        n = (a + b) / 2;

        if (Math.Abs(n) < nMinimoParaNoSerCero)
        {
            declinacionInicial = -Math.Log(produccionFinal / produccionInicial) / (tf);
            n = 0;
        }
        else
        {
            declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) / (n
* tf);
        }
        DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
        HallarR2(vectorDeDatos,    vectorLocalValoresDelModelo,    usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos,    vectorLocalValoresDelModelo,    usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;
    }

    public void ConstruirModeloHiperbolicoAutomaticoConNewtonRaphson(double[]
vectorDeDatos, out double[] vectorValoresDelModelo, int usarDatosDesde, int
usarDatosHasta, int numeroDeIteraciones, out double declinacionInicial, out double n, out
double r2, out double aic)
    {
        /*Se plantea que el modelo esté obligado a pasar por el punto inicial y final de los
datos
        */

        double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], ti = usarDatosDesde, tf = usarDatosHasta -
usarDatosDesde;
        double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
        double r2Derivada, paso = 0.1, a, b;
        int numeroDeParametros = 2;
        n = 0.33;
        declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) / (n *
tf);
        DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);

```

```

HallarR2Derivada(vectorDeDatos,          vectorLocalValoresDelModelo,
usarDatosDesde, usarDatosHasta, out r2Derivada, n, declinacionInicial);

if (r2Derivada != 0)
{
    if (r2Derivada < 0)
    {
        while (r2Derivada < 0)
        {
            n = n - paso;
            declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n))
/ (n * tf);
            DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
            HallarR2Derivada(vectorDeDatos,          vectorLocalValoresDelModelo,
usarDatosDesde, usarDatosHasta, out r2Derivada, n, declinacionInicial);
        }

        n = n + paso;
    }
    else
    {
        while (r2Derivada > 0)
        {
            n = n + paso;
            declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n))
/ (n * tf);
            DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
            HallarR2Derivada(vectorDeDatos,          vectorLocalValoresDelModelo,
usarDatosDesde, usarDatosHasta, out r2Derivada, n, declinacionInicial);
        }
    }

    a = n - paso;
    b = n;
    n = (a + b) / 2;
    for (int i = 0; i <= numeroDeliteraciones; i++)
    {

```

```

        declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) /
(n * tf);
        DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
        HallarR2Derivada(vectorDeDatos,          vectorLocalValoresDelModelo,
usarDatosDesde, usarDatosHasta, out r2Derivada, n, declinacionInicial);

        if(r2Derivada > 0 )
        {
            a = n;
        }
        else
        {
            b = n;
        }
        n = (a + b) / 2;
        declinacionInicial = (-1 + Math.Pow(produccionInicial / produccionFinal, n)) /
(n * tf);
    }
}

        DarValoresAlModeloHiperbolico(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial, n);
        HallarR2(vectorDeDatos,  vectorLocalValoresDelModelo,  usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos,  vectorLocalValoresDelModelo,  usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;
    }

    protected          void          DarValoresAlModeloHiperbolico(ref          double[]
vectorLocalValoresDelModelo, double produccionInicial, double declinacionInicial, double
n)
    {
        if(Math.Abs(n) < nMinimoParaNoSerCero)
        {
            DarValoresAlModeloExponencial(ref          vectorLocalValoresDelModelo,
produccionInicial, declinacionInicial);
        }
        else
        {
            for (int tiempo = 0; tiempo < vectorLocalValoresDelModelo.Length; tiempo++)

```

```

        {
            vectorLocalValoresDelModelo[tiempo] = produccionInicial * Math.Pow(1 + n
* declinacionInicial * tiempo, (-1 / n));
        }
    }
}

public void PronosticarConModeloHiperbolico(ref double[] vectorValoresDelModelo,
int tiempoAPartirDelCualSePronostica, double produccionInicial, double
declinacionInicial, double n)
{
    int tiempo;
    for (int i = 0; i < vectorValoresDelModelo.Length; i++)
    {
        tiempo = i + tiempoAPartirDelCualSePronostica;
        vectorValoresDelModelo[i] = produccionInicial * Math.Pow(1 + n *
declinacionInicial * tiempo, (-1 / n));
    }
}

protected void HallarR2(double[] vectorDeDatos, double[]
vectorLocalValoresDelModelo, int usarDatosDesde, int usarDatosHasta, out double r2)
{
    double suma = 0, varianza = 1;
    for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
    {
        suma = suma + Math.Pow((vectorLocalValoresDelModelo[i] - vectorDeDatos[i
+ usarDatosDesde]), 2);
    }
    HallarVarianzaDeDatos(vectorDeDatos, usarDatosDesde, usarDatosHasta, out
varianza);
    r2 = 1 - (suma /* / vectorLocalValoresDelModelo.Length */) / varianza;
}

protected void HallarR2Derivada(double[] vectorDeDatos, double[]
vectorLocalValoresDelModelo, int usarDatosDesde, int usarDatosHasta, out double
r2derivada, double n, double di)
{
    int tiempo;
    double suma = 0, varianza = 1, media, produccionInicial =
vectorDeDatos[usarDatosDesde];
    HallarMedia(vectorDeDatos, out media, usarDatosDesde, usarDatosHasta);
}

```

```

for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
{
    tiempo = i;
    suma = suma + (produccionInicial * Math.Pow((1 + n*di*tiempo), (-1/n)) -
vectorDeDatos[usarDatosDesde+ i]) * Math.Pow((1 + n * di * tiempo), (-1 / n)) * ((1 /
Math.Pow(n,2)) *Math.Log(1 + n * di * tiempo) - (1 / n)*((di * tiempo)/(1 + n * di * tiempo)));
}
HallarVarianzaDeDatos(vectorDeDatos, usarDatosDesde, usarDatosHasta, out
varianza);
r2derivada = 2*produccionInicial*(suma /* / vectorLocalValoresDelModelo.Length
*/) / (varianza);
}

```

```

protected void HallarVarianzaDeDatos(double[] vectorDeDatos, int
usarDatosDesde, int usarDatosHasta,out double varianza)
{
    double media, suma=0;
    HallarMedia(vectorDeDatos, out media, usarDatosDesde, usarDatosHasta);
    for (int i = usarDatosDesde; i <= usarDatosHasta; i++)
    {
        suma = suma + Math.Pow(vectorDeDatos[i] - media, 2);
    }
    varianza = suma /* / (usarDatosHasta - usarDatosDesde + 1)*/;
}

```

```

protected void HallarMedia(double[] vectorDatos, out double media, int
usarDatosDesde, int usarDatosHasta)
{
    double suma = 0;
    for (int i = usarDatosDesde; i <= usarDatosHasta; i++)
    {
        suma = suma + vectorDatos[i];
    }
    media = suma / (usarDatosHasta - usarDatosDesde + 1);
}

```

```

protected void HallarAIC(double[] vectorDeDatos, double[]
vectorLocalValoresDelModelo, int usarDatosDesde, int usarDatosHasta, out double aic,
int numeroDeParametros)
{
    double suma = 0;

```

```

    for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
    {
        suma = suma + Math.Pow((vectorLocalValoresDelModelo[i] - vectorDeDatos[i
+ usarDatosDesde]), 2);
    }
    aic = Math.Log(Math.Exp(2 * numeroDeParametros /
vectorLocalValoresDelModelo.Length) * suma / vectorLocalValoresDelModelo.Length);
}

```

```

public void ConstruirModeloExponencialQuePasePorEIUltimoPunto(double[]
vectorDeDatos, out double[] vectorValoresDelModelo, int usarDatosDesde, int
usarDatosHasta, out double declinacionInicial, ref double r2, ref double aic)
{
    int numeroDeParametros = 1;
    int tf = usarDatosHasta - usarDatosDesde, ti = usarDatosDesde;
    double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], declinacionInicialLocal;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

```

//Es necesario asegurarse de que el ultimo dato no sea cero, para que no se genere error al realizar el modelo

```

    if(vectorDeDatos[tf] != 0)
    {
        declinacionInicialLocal = -Math.Log(produccionFinal / produccionInicial) / (tf);

        DarValoresAlModeloExponencial(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicialLocal);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;
        declinacionInicial = declinacionInicialLocal;
    }
    else
    {

```

```

        MessageBox.Show("El ultimo dato de produccion es cero (0), por tanto no
        puede ser usado para realizar el modelo\nPor favor regrese a la pestaña de datos e ignore
        los ultimos datos de manera que el ultimo a usar sea diferente de cero (0)", "Error de
        datos", MessageBoxButtons.OK, MessageBoxIcon.Error);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;
        declinacionInicial = 0;
    }
}

```

```

}

public void ConstruirModeloExponencialEspecificandoDi(double[] vectorDeDatos,
out double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, double
declinacionInicial, ref double r2, ref double aic)
{
    int numeroDeParametros = 1;
    int tf = usarDatosHasta - usarDatosDesde, ti = usarDatosDesde;
    double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], declinacionInicialLocal;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

    declinacionInicialLocal = declinacionInicial;

    DarValoresAlModeloExponencial(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicialLocal);
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
    HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}
}

```

```

public void ConstruirMejorModeloExponencial(REngine motorDeR, string
nombreDelVectorDeDatosR, NumericVector datosParaR, double[] vectorDeDatos, out
double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, out double
declinacionInicial, ref double produccionInicial, ref double r2, ref double aic)
{
    int numeroDeParametros = 2;
    double declinacionInicialLocal;
}

```

```

    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

    NumericVector coeficientesDesdeR = motorDeR.CreateNumericVector(2);

    int numeroDeDatos = usarDatosHasta - usarDatosDesde + 1;

    motorDeR.Evaluate("matriz <- matrix(nrow = length(" +
nombreDelVectorDeDatosR + "), ncol = 2)");
    motorDeR.Evaluate("tiempo <- c(1:(length(" + nombreDelVectorDeDatosR +
"))));");
    motorDeR.Evaluate("matriz[,1] <- 1");
    motorDeR.Evaluate("matriz[,2] <- tiempo");
    motorDeR.Evaluate("vectorLogaritmico <- log(" + nombreDelVectorDeDatosR +
)");
    motorDeR.Evaluate("i=1\nwhile (i < length(vectorLogaritmico))\n{\nif (" +
nombreDelVectorDeDatosR + "[i] == 0)\n{\nvectorLogaritmico[i] = 0\n}\ni = i + 1\n}");
    motorDeR.Evaluate("coeficientesLogaritmico <- solve(t(matriz) %*% matriz) %*%
(t(matriz) %*% vectorLogaritmico)");

    coeficientesDesdeR =
motorDeR.Evaluate("coeficientesLogaritmico").AsNumeric();
    produccionInicial = Math.Exp(coeficientesDesdeR[0]);
    declinacionInicialLocal = -coeficientesDesdeR[1];
    DarValoresAlModeloExponencial(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicialLocal);
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
    HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);

    vectorValoresDelModelo = vectorLocalValoresDelModelo;
    declinacionInicial = declinacionInicialLocal;

}

protected void DarValoresAlModeloExponencial(ref double[]
vectorLocalValoresDelModelo, double produccionInicial, double declinacionInicial)
{
    for (int tiempo = 0; tiempo < vectorLocalValoresDelModelo.Length; tiempo++)
    {

```

```

        vectorLocalValoresDelModelo[tiempo] = produccionInicial * Math.Exp(-
declinacionInicial*tiempo);
    }

}

public void PronosticarConModeloExponencial(ref double[]
vectorValoresDelModelo, int tiempoAPartirDelCualSePronostica, double
produccionInicial, double declinacionInicial)
{
    int tiempo;
    for (int i = 0; i < vectorValoresDelModelo.Length; i++)
    {
        tiempo = i + tiempoAPartirDelCualSePronostica;
        vectorValoresDelModelo[i] = produccionInicial * Math.Exp(-declinacionInicial *
tiempo);
    }
}

```

```

public void ConstruirModeloArmonicoQuePasePorElUltimoPunto(double[]
vectorDeDatos, out double[] vectorValoresDelModelo, int usarDatosDesde, int
usarDatosHasta, out double declinacionInicial,ref double r2,ref double aic)
{
    int numeroDeParametros = 1;
    int tf = usarDatosHasta - usarDatosDesde, ti = usarDatosDesde;
    double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], declinacionInicialLocal;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

```

//Es necesario asegurarse de que el ultimo dato no sea cero, para que no se genere error al realizar el modelo

```

    if (vectorDeDatos[tf] != 0)
    {
        declinacionInicialLocal = ((produccionInicial / produccionFinal) - 1) / tf;

        DarValoresAlModeloArmonico(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicialLocal);

```

```

        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;

        declinacionInicial = declinacionInicialLocal;
    }
    else
    {
        MessageBox.Show("El ultimo dato de produccion es cero (0), por tanto no
puede ser usado para realizar el modelo\nPor favor regrese a la pestaña de datos e ignore
los ultimos datos de manera que el ultimo a usar sea diferente de cero (0)", "Error de
datos", MessageBoxButtons.OK, MessageBoxIcon.Error);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;
        declinacionInicial = 0;
    }
}
}

```

```

public void ConstruirModeloArmonicoEspecificandoDi(double[] vectorDeDatos, out
double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, double
declinacionInicial, ref double r2, ref double aic)

```

```

{
    int numeroDeParametros = 1;
    int tf = usarDatosHasta - usarDatosDesde, ti = usarDatosDesde;
    double produccionInicial = vectorDeDatos[usarDatosDesde], produccionFinal =
vectorDeDatos[usarDatosHasta], declinacionInicialLocal;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

```

//Es necesario asegurarse de que el ultimo dato no sea cero, para que no se genere error al realizar el modelo

```

        declinacionInicialLocal = declinacionInicial;
        DarValoresAlModeloArmonico(ref vectorLocalValoresDelModelo,
produccionInicial, declinacionInicialLocal);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);
        vectorValoresDelModelo = vectorLocalValoresDelModelo;

```

```

    }

    public void ConstruirMejorModeloArmonico(ref REngine motorDeR, string
nombreDelVectorDeDatosR, NumericVector datosParaR, double[] vectorDeDatos, out
double[] vectorValoresDelModelo, int usarDatosDesde, int usarDatosHasta, out double
declinacionInicial, ref double produccionInicial, ref double r2,ref double aic)
    {
        int numeroDeParametros = 2;
        double declinacionInicialLocal;
        double produccionInicialLocal;
        double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

        NumericVector coeficientesDesdeR = motorDeR.CreateNumericVector(2);

        int numeroDeDatos = usarDatosHasta - usarDatosDesde + 1;
        motorDeR.Evaluate("matriz <- matrix(nrow = length(" +
nombreDelVectorDeDatosR + "), ncol = 2)");
        motorDeR.Evaluate("tiempo <- c(1:(length(" + nombreDelVectorDeDatosR +
"))));");
        motorDeR.Evaluate("matriz[,1] <- 1");
        motorDeR.Evaluate("matriz[,2] <- tiempo");
        motorDeR.Evaluate("vectorInverso <- 1/(" + nombreDelVectorDeDatosR + ")");
        motorDeR.Evaluate("i=1\nwhile (i < length(vectorInverso))\n{\nif (" +
nombreDelVectorDeDatosR + "[i] == 0)\n{\nvectorInverso[i] = 0\n}\ni = i + 1\n}");
        motorDeR.Evaluate("coeficientesArmonico <- solve(t(matriz) %*% matriz) %*%
(t(matriz) %*% vectorInverso)");

        coeficientesDesdeR = motorDeR.Evaluate("coeficientesArmonico").AsNumeric();
        produccionInicialLocal = 1/(coeficientesDesdeR[0]);
        declinacionInicialLocal = coeficientesDesdeR[1] / coeficientesDesdeR[0];
        DarValoresAlModeloArmonico(ref vectorLocalValoresDelModelo,
produccionInicialLocal, declinacionInicialLocal);
        HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
        HallarAIC(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out aic, numeroDeParametros);

        vectorValoresDelModelo = vectorLocalValoresDelModelo;
        produccionInicial = produccionInicialLocal;
        declinacionInicial = declinacionInicialLocal;

```

```

    }

    protected void DarValoresAlModeloArmonico(ref double[]
vectorLocalValoresDelModelo, double produccionInicial, double declinacionInicial)
    {
        vectorLocalValoresDelModelo[0] = produccionInicial;
        for (int tiempo = 1; tiempo < vectorLocalValoresDelModelo.Length; tiempo++)
        {
            vectorLocalValoresDelModelo[tiempo] = produccionInicial / (1 +
declinacionInicial * tiempo);
        }
    }

    public void PronosticarConModeloArmonico(ref double[] vectorValoresDelModelo,
int tiempoAPartirDelCualSePronostica, double produccionInicial, double
declinacionInicial)
    {
        int tiempo;
        if (tiempoAPartirDelCualSePronostica == 0)
        {
            vectorValoresDelModelo[0] = produccionInicial;
            for (int i = 1; i < vectorValoresDelModelo.Length; i++)
            {
                tiempo = i;
                vectorValoresDelModelo[i] = produccionInicial / (1 + declinacionInicial *
tiempo);
            }
        }
        else
        {
            for (int i = 0; i < vectorValoresDelModelo.Length; i++)
            {
                tiempo = tiempoAPartirDelCualSePronostica + i;
                vectorValoresDelModelo[i] = produccionInicial / (1 + declinacionInicial *
tiempo);
            }
        }
    }
}

```

```

public void ConstruirMejorModeloArima(ref REngine motorDeR, string
nombreDelVectorDeDatosR,out string modeloArima, NumericVector datosParaR,
double[] vectorDeDatos, out double[] vectorValoresDelModelo, out NumericVector
coeficientesArima, int usarDatosDesde, int usarDatosHasta, out double valorP, out
double aic, out double r2)
{
    int numeroDeParametros;
    NumericVector valoresDelModelo;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
    motorDeR.Evaluate("modeloArima <- auto.arima(" + nombreDelVectorDeDatosR
+ ")");
    motorDeR.Evaluate("arimaParaPronostico <- forecast(modeloArima)");
    numeroDeParametros =
Convert.ToInt32(motorDeR.Evaluate("length(modeloArima$coef)").AsNumeric().First());
    modeloArima =
motorDeR.Evaluate("arimaParaPronostico$method").AsCharacter().ToArray().First().Su
bstring(0,12);
    motorDeR.Evaluate("residuos <- arimaParaPronostico$residuals");
    motorDeR.Evaluate("testDeBox <- Box.test(residuos, lag = 12, type = c(\"Ljung-
Box\"), fitdf = 0)");
    valorP =
Convert.ToDouble(motorDeR.Evaluate("testDeBox$p.value").AsNumeric().First());
    valoresDelModelo = motorDeR.Evaluate(nombreDelVectorDeDatosR + "-
residuos").AsNumeric();
    coeficientesArima =
motorDeR.Evaluate("arimaParaPronostico$model$coef").AsNumeric();

    aic =
Convert.ToInt32(motorDeR.Evaluate("arimaParaPronostico$model$aic").AsNumeric().Fi
rst());

    for (int i=0;i< vectorLocalValoresDelModelo.Length; i++)
    {
        vectorLocalValoresDelModelo[i] = valoresDelModelo[i];
    }
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}

```

```

public void ConstruirModeloArimaEspecificado(ref REngine motorDeR, string
nombreDelVectorDeDatosR, out string modeloArima, NumericVector datosParaR,
double[] vectorDeDatos, out double[] vectorValoresDelModelo, out NumericVector
coeficientesArima, int usarDatosDesde, int usarDatosHasta, out double valorP, out
double aic, out double r2, int p, int d, int q)
{
    int numeroDeParametros;
    NumericVector valoresDelModelo;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
    motorDeR.Evaluate("modeloArima <- arima(" + nombreDelVectorDeDatosR + ",
order = c("+p+", "+d+", "+q+")");

    numeroDeParametros =
Convert.ToInt32(motorDeR.Evaluate("length(modeloArima$coef)").AsNumeric().First());
    modeloArima = "ARIMA(" + p + ", " + d + ", " + q + ")";

    aic =
Convert.ToDouble(motorDeR.Evaluate("modeloArima$aic").AsNumeric().First());
    motorDeR.Evaluate("residuos <- modeloArima$residuals");
    motorDeR.Evaluate("testDeBox <- Box.test(residuos, lag = 12, type = c(\"Ljung-
Box\"), fitdf = 0)");
    valorP =
Convert.ToDouble(motorDeR.Evaluate("testDeBox$p.value").AsNumeric().First());
    valoresDelModelo = motorDeR.Evaluate(nombreDelVectorDeDatosR + "-
residuos").AsNumeric();
    coeficientesArima = motorDeR.Evaluate("modeloArima$coef").AsNumeric();

    for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
    {
        vectorLocalValoresDelModelo[i] = valoresDelModelo[i];
    }
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}

public void ConstruirArimaConMejorAic(string direccionFuncion, ref REngine
motorDeR, string nombreDelVectorDeDatosR, int d, out string modeloArima,
NumericVector datosParaR, double[] vectorDeDatos, out double[]

```

```

vectorValoresDelModelo, out NumericVector coeficientesArima, int usarDatosDesde, int
usarDatosHasta, out double valorP, out double aic, out double r2)
{
    int numeroDeParametros, p, q;
    NumericVector valoresDelModelo;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];

    string direccionFuncionParaR = CambiarCaracter(direccionFuncion, @"\", "/");

    motorDeR.Evaluate("source(\""+direccionFuncionParaR+"\")");
    motorDeR.Evaluate("resultadoMejorARIMA <- MejorARIMA(" +
nombreDelVectorDeDatosR + ", "+ d +")");
    motorDeR.Evaluate("modeloArima <- resultadoMejorARIMA$modelo");

    numeroDeParametros =
Convert.ToInt32(motorDeR.Evaluate("length(modeloArima$coef)").AsNumeric().First());
    p =
Convert.ToInt32(motorDeR.Evaluate("resultadoMejorARIMA$p").AsNumeric().First());
    q =
Convert.ToInt32(motorDeR.Evaluate("resultadoMejorARIMA$q").AsNumeric().First());
    modeloArima = "ARIMA(" + p + ", " + d + ", " + q + ")";

    aic =
Convert.ToDouble(motorDeR.Evaluate("modeloArima$aic").AsNumeric().First());
    motorDeR.Evaluate("residuos <- modeloArima$residuals");
    motorDeR.Evaluate("testDeBox <- Box.test(residuos, lag = 12, type = c(\"Ljung-
Box\"), fitdf = 0)");
    valorP =
Convert.ToDouble(motorDeR.Evaluate("testDeBox$p.value").AsNumeric().First());
    valoresDelModelo = motorDeR.Evaluate(nombreDelVectorDeDatosR + "-
residuos").AsNumeric();
    coeficientesArima = motorDeR.Evaluate("modeloArima$coef").AsNumeric();

    for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
    {
        vectorLocalValoresDelModelo[i] = valoresDelModelo[i];
    }
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
}

```

```

    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}

protected string CambiarCaracter(string original, string aSustituir, string sustituto)
{
    string izq, der;
    int contador = 0;
    int pos = original.IndexOf(aSustituir);

    if (aSustituir.Length != 0)
    {
        while (pos >= contador)
        {
            izq = original.Substring(0, pos);
            der = original.Substring(pos + aSustituir.Length, original.Length - pos -
aSustituir.Length);
            original = izq + sustituto + der;
            contador = izq.Length + sustituto.Length;
            pos = original.IndexOf(aSustituir);
        }
    }
    return original;
}

```

```

public void ConstruirModeloCombinado(ref REngine motorDeR, out string
modeloArima, double[] vectorDeDatos, out double[] vectorValoresDelModelo, out
NumericVector coeficientesArima, int usarDatosDesde, int usarDatosHasta, out double
declinacionInicial, out double n, out double r2, out double aic, out double valorP)
{
    string nombreDelVectorDeDatosR = "datosParaArima";
    int numeroDelteraciones = 80;
    double[] vectorModeloHiperbolico;
    ConstruirModeloHiperbolicoAutomatico(vectorDeDatos, out
vectorModeloHiperbolico, usarDatosDesde, usarDatosHasta, numeroDelteraciones, out
declinacionInicial, out n, out r2, out aic);
    NumericVector datosParaArima =
motorDeR.CreateNumericVector(usarDatosHasta - usarDatosDesde + 1);
    for(int i=0; i<datosParaArima.Length; i++)
    {

```

```

        datosParaArima[i] = vectorDeDatos[i] + usarDatosDesde] -
vectorModeloHiperbolico[i];
    }
    motorDeR.SetSymbol(nombreDelVectorDeDatosR, datosParaArima);

    NumericVector valoresDelModeloParteArima;
    double[] vectorLocalValoresDelModelo = new double[usarDatosHasta -
usarDatosDesde + 1];
    motorDeR.Evaluate("modeloParteArima <- auto.arima(" +
nombreDelVectorDeDatosR + ")");
    motorDeR.Evaluate("arimaDatos <- forecast(modeloParteArima)");
    modeloArima =
motorDeR.Evaluate("arimaDatos$method").AsCharacter().ToArray().First().Substring(0,
12);
    motorDeR.Evaluate("residuosParteArima <- arimaDatos$residuals");
    motorDeR.Evaluate("testDeBoxParteArima <- Box.test(residuosParteArima, lag =
12, type = c(\"Ljung-Box\"), fitdf = 0)");
    valorP =
Convert.ToDouble(motorDeR.Evaluate("testDeBoxParteArima$p.value").AsNumeric().Fi
rst());
    valoresDelModeloParteArima = motorDeR.Evaluate(nombreDelVectorDeDatosR
+ "- residuosParteArima").AsNumeric();
    coeficientesArima =
motorDeR.Evaluate("arimaDatos$model$coef").AsNumeric();

    aic =
Convert.ToInt32(motorDeR.Evaluate("arimaDatos$model$aic").AsNumeric().First());

    for (int i = 0; i < vectorLocalValoresDelModelo.Length; i++)
    {
        vectorLocalValoresDelModelo[i] = vectorModeloHiperbolico[i] +
valoresDelModeloParteArima[i];
    }
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
    HallarR2(vectorDeDatos, vectorLocalValoresDelModelo, usarDatosDesde,
usarDatosHasta, out r2);
    vectorValoresDelModelo = vectorLocalValoresDelModelo;
}

public void PronosticarConModeloCombinado(bool pronosticoSolo, bool
pronosticoTodosLosDatos, int j, REngine motorDeR, string

```

```

nombreDelVectorDeDatosEnR, ref double[] vectorPronostico, double[]
vectorValoresDelModelo,int usarDatosDesde, int usarDatosHasta, int
intervalosDeTiempoAPronosticar, int tiempoAPartirDelCualSePronostica, double
produccionInicial, double declinacionInicial, double n)
{
    double[] vectorLocalHiperbólico = new double[vectorPronostico.Length];
    int tiempo;
    for (int i = 0; i < vectorLocalHiperbólico.Length; i++)
    {
        tiempo = i + tiempoAPartirDelCualSePronostica;
        vectorLocalHiperbólico[i] = produccionInicial * Math.Pow(1 + n *
declinacionInicial * tiempo, (-1 / n));
    }

    NumericVector prediccionParteArima;
    motorDeR.Evaluate("pronostico <- forecast(modeloParteArima," +
intervalosDeTiempoAPronosticar + ")");
    prediccionParteArima = motorDeR.Evaluate("pronostico$mean").AsNumeric();

    if (pronosticoSolo)
    {
        for (int i = 0; i < vectorLocalHiperbólico.Length; i++)
        {
            vectorPronostico[i] = vectorLocalHiperbólico[i] + prediccionParteArima[i];
        }
    }
    else
    {
        if (pronosticoTodosLosDatos)
        {
            for (int i = 0; i < usarDatosHasta - usarDatosDesde + 1; i++)
            {
                vectorPronostico[i] = vectorValoresDelModelo[i];
            }
            for (int i = 0; i < intervalosDeTiempoAPronosticar; i++)
            {
                vectorPronostico[i + usarDatosHasta - usarDatosDesde + 1] =
vectorLocalHiperbólico[i + usarDatosHasta - usarDatosDesde + 1] +
prediccionParteArima[i]; ;
            }
        }
        else
    }
}

```

```

        {
            for (int i = 0; i < j; i++)
            {
                vectorPronostico[j - 1 - i] =
vectorValoresDelModelo[vectorValoresDelModelo.Length - 1 - i];
            }

            for (int i = 0; i < intervalosDeTiempoAPronosticar; i++)
            {
                vectorPronostico[i + j] = vectorLocalHiperbólico[i + j] +
prediccionParteArima[i];
            }
        }
    }

    for (int i = 0; i < vectorLocalHiperbólico.Length; i++)
    {
        if(vectorPronostico[i] < 0)
        {
            vectorPronostico[i] = 0;
        }
    }
}
}

```

```

public void ConstruirMejorArimaConAicMetodoUtilParaHilos(ref REngine motorDeR,
string nombreDelVectorDeDatosR, out string modeloArima, NumericVector datosParaR,
double[] vectorDeDatos, out double[] vectorValoresDelModelo, out NumericVector
coeficientesArima, int usarDatosDesde, int usarDatosHasta, out double valorP, out
double aic, out double r2)
{
    double[,] matrizAic = new double[10, 10];
    string nombreDelModeloEnR = "modelo";
    MessageBox.Show("Este proceso puede tardar varios minutos", "Continuar...",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    int p = 1, q = 1, d = 1;

    List<Thread> threads = new List<Thread>();

    while (p <= 10)
    {

```

```

        while (q <= 10)
        {
            motorDeR.Evaluate("'" + nombreDelModeloEnR + "' <- auto.arima(" +
nombreDelVectorDeDatosR + ", max.p = " + p + ", max.q = " + q + ", start.p = " + p + ",
start.q = " + q + ", d=" + d + ")");
            matrizAic[p-1, q-1] =
Convert.ToDouble(motorDeR.Evaluate(nombreDelModeloEnR +
"$aic").AsNumeric().First());
            q++;
        }
        p++;
    }
    p = 1;
    q = 2;
    int pMejorAic = p;
    int qMejorAic = q;
    double mejorAic = matrizAic[0, 0];
    while (p <= 10)
    {
        while (q <= 10)
        {
            if(matrizAic[p-1, q-1] < mejorAic)
            {
                pMejorAic = p;
                qMejorAic = q;
                mejorAic = matrizAic[p-1, q-1];
            }
        }
    }
    ConstruirModeloArimaEspecificado(ref motorDeR, nombreDelVectorDeDatosR,
out modeloArima, datosParaR, vectorDeDatos, out vectorValoresDelModelo, out
coeficientesArima, usarDatosDesde, usarDatosHasta, out valorP, out aic, out r2, p, d, q);
}
}
}

```