UNIVERSIDAD TORCUATO DI TELLA

MASTER IN MANAGEMENT + ANALYTICS

REDES NEURONALES RECURRENTES APLICADAS A SERIES DE TIEMPO: PREDICCIÓN DE PASAJEROS DE RUTAS REGIONALES EN ARGENTINA

TESIS

Constanza Gaset

Mayo 2022

Tutor: Magdalena Cornejo

Resumen

Para la mayoría de las empresas es importante contar con información actualizada y precisa, sobre todo en contextos de gran incertidumbre. En la industria aeronáutica, particularmente, una predicción precisa del número de pasajeros resulta vital ya que proporciona información valiosa para decisiones de planificación de la oferta de vuelos y estrategias de gestión de ingresos o *revenue management* que afectan no solo al nivel de servicio sino también a la maximización de los ingresos de las aerolíneas. Asimismo, el contexto reciente con la Pandemia COVID-19, afectó drásticamente al sector aerocomercial generando un nivel adicional de dificultad a la tarea de predecir el número de pasajeros futuros. Sin embargo, nos brindó un marco muy particular para evaluar el desempeño de distintos modelos predictivos en un contexto de gran incertidumbre y quiebre en las series a pronosticar. Este estudio propone explorar diferentes modelos complejos para predecir los pasajeros de las principales rutas regionales operadas en Argentina, a nivel origen-destino con un horizonte a doce meses. Se busca entender si éstos logran incorporar mejor el contexto reciente y generar predicciones confiables, como así también brindar información de utilidad a las aerolíneas. Se exploran modelos basados en *Deep Learning* como las Redes Neuronales Recurrentes (RNN), particularmente las redes de memoria a corto-largo plazo (LSTM, por sus siglas en inglés). Se aplican redes univariadas y multivariadas comparando los resultados obtenidos con modelos complejos clásicos como TBATS, utilizando como medida de comparación entre modelos a la Raíz Cuadrada del Error Cuadrático Medio (RMSE, por sus siglas en ingles). Las redes LSTM demuestran un desempeño superior sobre los modelos clásicos, pero aun con errores promedios altos relacionados al contexto cambiante de la industria. Finalmente, se realiza un caso simulado de aplicación de negocio con los resultados obtenidos en las predicciones, explicando sus potenciales usos en la toma de decisiones de una aerolínea.

Abstract

For most companies, it is important to have updated and accurate information, especially in contexts of great uncertainty. In the aeronautical industry, an accurate prediction of the number of passengers is key to provide valuable information for flight supply planning and revenue management strategies that affect not only the service level but also the airline revenue maximization. COVID-19 Pandemic recent context drastically affected the commercial aviation sector generating an additional level of difficulty in predicting the number of future passenger's task. This study proposes to explore different complex models to predict the passengers of the main regional routes operated in Argentina, at the origin-destination level with a horizon of twelve months. It seeks to understand if they can incorporate recent context and generate reliable predictions, as well as provide useful information to airlines. Deep Learning-based models such as Recurrent Neural Networks (RNN), particularly Long-Short-Term Memory (LSTM) networks, are explored. Univariate and multivariate networks are applied to compare the results obtained with classical complex models such as TBATS, using the Square Root Mean Square Error (RMSE) as a comparison measure between models. LSTM networks demonstrate superior performance over classic models, but still with high average errors related to the changing context of the industry. Finally, a simulated case of business application is developed with the predictions explaining their potential uses in the decision-making of an airline.

Índice

Índice de Tablas

Índice de Figuras

Índice de Ecuaciones

1. Introducción

Para la mayoría de las empresas es importante contar con información actualizada y lo más precisa posible, sobre todo en contextos de gran incertidumbre. En la industria aeronáutica, particularmente, resulta vital poder contar con información sobre la demanda futura de vuelos, de modo de planificar la oferta y establecer estrategias de *revenue management* que optimicen la flota y los asientos a vender.

Existen diversos sistemas que contratan las aerolíneas que les proporcionan dicha información, considerando datos privados de sus ventas y otras variables que puedan afectar a la demanda (competencia, disponibilidad de asientos propios y de la competencia, entre otras).

Actualmente no se cuenta con proyecciones agregadas sobre cantidad de pasajeros futuros por parte de organismos públicos. La Administración Nacional de Aviación Civil (ANAC) publica mensualmente los datos de pasajeros comerciales volados a nivel origen-destino (OD) y aeropuerto, pero no cuenta con predicciones propias respecto a estas variables.

Por otro lado, en los últimos años, junto con el desarrollo de *machine* y *deep learning*, surgieron técnicas complejas y modelos alternativos para abordar los problemas de predicción de series de tiempo, que en algunos casos superan a los métodos estadísticos tradicionales. Las redes neuronales (NN) toman especial relevancia en este aspecto, dado que son capaces de modelar y predecir series de tiempo lineales y no lineales con un buen grado de precisión, capturando cualquier tipo de interrelación entre los datos. Estas pueden predecir valores futuros en un paso o realizar predicciones recursivas de largo plazo utilizando los propios valores estimados para las proyecciones. En este punto es que aparecen las redes neuronales recurrentes (RNN), un tipo de redes con una arquitectura que implementa una cierta memoria y, por lo tanto, un sentido temporal a las estimaciones.

Por ello, esta tesis tiene por principal objetivo desarrollar modelos de predicción de pasajeros comerciales para las principales rutas regionales operadas en Argentina, a nivel origen-destino con un horizonte temporal de un año (12 meses) adelante. Asimismo, busca contrastar los resultados de diferentes modelos univariados y multivariados, como redes neuronales recurrentes y modelos estadísticos clásicos y entender como estas estimaciones incorporan el contexto reciente de pandemia que afectó drásticamente la circulación de pasajeros durante el 2020.

Maximizar los ingresos por cada asiento vendido, esto es, *vender el asiento correcto, en el momento correcto y al cliente correcto* es la esencia del *revenue* en cualquier aerolínea. Es por ello que poder contar con información lo más precisa posible respecto de esa demanda puede generar un impacto importante en los ingresos de éstas.

A su vez, poder brindar información pública respecto de la demanda o tráfico esperado de pasajeros a nivel ruta puede serles de gran utilidad, no solo a las aerolíneas que ya se encuentran operando esas rutas, sino a cualquier aerolínea que considere entrar al mercado. Aplicando un factor de *market share* sobre la estimación brindada, cualquier aerolínea podrá incorporar ese input dentro de sus estimaciones o bien como input de cualquier análisis sobre el mercado. Esta noción de *market size*, ponderada por el share que una aerolínea apunta a capturar, resulta una base fundamental para definir sus estrategias de negocio y, a su vez, aumentar la precisión de éstas.

La presente tesis se estructura de la siguiente manera. En la sección 1 se desarrolla una breve introducción y contexto, destacando algunos trabajos relacionados. Luego se define el problema junto a la justificación y alcance del presente análisis. Se describe el objetivo general de la tesis, detallando asimismo los objetivos específicos. En la sección 2 se describen los datos y el *dataset* que se utiliza en la tesis. Asimismo, se realiza un análisis exploratorio de los datos, analizando componentes de tendencia y estacionalidad de las series temporales utilizadas. En la sección 3 se explica la metodología utilizada, empezando con una introducción a pronósticos de series de tiempo para pasar luego a modelos de series temporales y redes neuronales (redes neuronales recurrentes y modelos *Long-Short Term Memory*). Luego se desarrollan los modelos de redes neuronales univariados y multivariados como así también modelos estadísticos tradicionales, explicando las transformaciones aplicadas al *dataset* para ser utilizado como input de cada uno de los modelos. También se abordan la implementación y entrenamiento de los modelos. En la sección 4 se exponen los resultados obtenidos con los modelos explorados en esta tesis junto a un caso simulado de aplicación de negocio. Por último, en la sección 5 se desarrollan las conclusiones y recomendaciones, contemplando limitaciones del presente trabajo y posibles líneas futuras de investigación.

1.1. Contexto

En las últimas décadas, las redes neuronales han sido utilizadas para pronosticar demanda y predecir situaciones en diversos campos. Estas han ido ganando espacio al demostrar su precisión en la predicción y capacidad para captar patrones complejos en las series de tiempo en comparación con los métodos tradicionales estadísticos. Weatherford (2003)[1] fue el primer estudio comparativo explorando las redes neuronales aplicadas a la industria aeronáutica, comparándolas con los métodos tradicionales como promedios móviles, suavizado exponencial, regresión, entre otros. Estos métodos fueron comparados sobre la base del error porcentual absoluto medio (MAPE, por sus siglas en inglés), donde las redes neuronales tuvieron una *performance* superior al resto de los métodos estudiados. Srisaeng, Baxter y Wild (2015)[2] aplicaron dos modelos de redes neuronales multicapa a la estimación de pasajeros y el ingreso por pasajero y por kilómetro (*Revenue Passager per Kilometers o RPKs por sus siglas en inglés)* en Australia, utilizando un set de diversas variables como Producto Bruto Interno (PBI), precio del combustible, tasa de desempleo, entre otras. Por su parte, Mohie El-Din (2017)[3] utilizó una red neuronal de retropropagación o *backpropagation* y un algoritmo genético para estimar la demanda de pasajeros nacionales e internacionales en Egipto.

En este camino es que aparecen las redes neuronales recurrentes como modelos superadores para predecir series de tiempo. Gupta, Sharma y Sangwan (2019)[4], exploraron el uso de Redes Neuronales Recurrentes, particularmente redes Long-Short Term Memory (LSTM) para predecir pasajeros de las aerolíneas. Estos resultados fueron comparados con los alcanzados por los esquemas existentes explorados por otros trabajos (algunos nombrados en este apartado),

[1] Lawrence R. Weatherford (2003), Neural Network forecasting for airlines: A comparative analysis, Journal of Revenue and Pricing Management, Volume 1, Issue 4, 2003, pp 319–331.

[2] Srisaeng, P., Baxter, G. and Wild, G. (2015), Using an artificial neural network approach to forecast Australia's domestic passenger air travel demand, World Review of Intermodal Transportation Research, Vol. 5, No. 3, 2015, pp 281-313

[3] M. M. Mohie El-Din, M. S. Farag and A. A. Abouzeid, Airline Passenger Forecasting in EGYPT (Domestic and International), International Journal of Computer Application (0975-8887), Vol. 165, Issue.6, May 2017.

[4] Gupta,V., Sharma, K. and Sangwan, M.S. (2019) AIRLINES PASSENGER FORECASTING USING LSTM BASED RECURRENT NEURAL NETWORKS, International Journal "Information Theories and Applications", Vol. 26, Number 2, 2019, pp 178-187.

utilizando MAPE como medida de error. Se corrobora que las redes LSTM se desempeñaron mejor que otros esquemas existentes para predecir pasajeros, presentando el menor error entre los modelos explorados.

Por último, Quang Hung Do, Shih-Kuei Lo, Jeng-Fung Chen, Chi-Luan Le y Luong Hoang Anh (2020)[5] realizaron un estudio donde compararon la performance de modelos SARIMA y redes LSTM para predecir los pasajeros del Aeropuerto Internacional de Inchenon. Se exploraron diversos criterios de performance entre MAPE, error cuadrático medio, la raíz del error cuadrático medio y la desviación media absoluta (MSE, RMSE y MAD, por sus siglas en inglés), y si bien ambos modelos mostraron buena *performance*, las redes *Long-Short Term Memory* (LSTM, por sus siglas en inglés) demostraron una *performance* superior al modelo SARIMA, con el menor error en la comparación de los modelos.

1.2. Problema

Para poder planificar la oferta de vuelos, definir estrategias de ventas y *revenue management*, las aerolíneas necesitan contar con información de calidad respecto a la demanda de pasajeros futura. Poder contar con información pública de esta demanda ayudaría a las aerolíneas a mejorar sus esfuerzos por estimar la demanda de pasajeros de sus vuelos e incluso podría brindar información sobre tamaño y dinámica del mercado a cualquier aerolínea que esté evaluando entrar en el mercado regional en Argentina.

Actualmente, el personal de Empresa Argentina de Navegación Aérea S.E. (EANA) de cada aeropuerto registra los aterrizajes y despegues que se llevan a cabo, detallando los pasajeros transportados, hora de salida/llegada, origen y destino, carga transportada, aerolínea u operador. A partir de estos datos, se construye una base de datos, consolidando la información de cada uno de los aeropuertos y nivel origen-destino (OD), con una actualización mensual. Esta base es administrada por la Administración Nacional de Aviación Civil (ANAC).

A partir del contexto reciente, en el que durante el año 2020 el sector aeronáutico fue golpeado por la pandemia COVID-19, las estimaciones de pasajeros se han vuelto una tarea aún más compleja. Es por esto, que además de considerar los patrones históricos de los datos de pasajeros, se deben considerar cuestiones de contexto que afectan actualmente la actividad y que son altamente inciertas de cara a los próximos años.

Justificación

Todas las aerolíneas que operen rutas regionales en Argentina tienen la necesidad de estimar el número de pasajeros para sus rutas a nivel OD, a fin generar una planificación eficiente de sus recursos.

El desarrollo de modelos y diferentes técnicas para la estimación de pasajeros futuros en rutas regionales de Argentina les brindará información útil para las aerolíneas, quienes podrán nutrirse de información pública que les permita complementar sus esfuerzos para predecir los pasajeros de sus vuelos.

[5] Quang Hung Do, Shih-Kuei Lo, Jeng-Fung Chen, Chi-Luan Le and Luong Hoang Anh (2020), *Forecasting Air Passenger Demand: A Comparison of LSTM and SARIMA*, Journal of Computer Science, Original Research Paper, 2020

Asimismo, el presente trabajo considerará datos recientes de contexto por la pandemia COVID-19 y tratará de estimar pasajeros considerando esto, evaluando la *performance* de los modelos buscando que los mismo capten estas particularidades.

Alcances

a) El presente estudio analizará los datos históricos publicados por ANAC en cada una de las rutas OD seleccionadas para el análisis.

b) Dentro del análisis se incluirán también otras variables que pueden explicar la demanda futura de pasajeros, tales como la cantidad de vuelos por ruta OD, el tipo de cambio y un indicador de la actividad económica de Argentina.

c) Se analizará la *performance* de métodos clásicos de pronóstico de series de tiempo con la de métodos más complejos, como redes neuronales recurrentes.

d) La finalidad del estudio es proporcionar a las aerolíneas una predicción agregada de pasajeros futuros a doce meses (12) o un año para las principales rutas regionales de Argentina, nivel OD.

1.3. Objetivo

El principal objetivo de esta tesis es generar predicciones a corto y mediano plazo (un año) de pasajeros comerciales para las principales rutas regionales operadas en Argentina, a nivel origen-destino. Para ello se utilizarán datos históricos de pasajeros y vuelos comerciales publicados por ANAC y datos externos que afecten a la demanda de pasajeros. Estas proyecciones serán de carácter público, pudiendo tener acceso cualquier aerolínea que lo desee.

Objetivos Específicos

- Explorar y analizar los datos disponibles, identificando principales componentes de las series de tiempo.
- Transformar y preparar los datos para la implementación de los modelos de pronóstico.
- Explorar y analizar técnicas de *machine learning*: redes neuronales recurrentes univariadas y multivariadas. Se utilizarán los modelos *Long-Short Term Memory*.
- Analizar modelos estadísticos clásicos: modelos TBATS.
- Comparar la *performance* de los métodos de pronóstico explorados y elegir el mejor modelo. Realizar estimaciones o predicciones a un año y realizar una aplicación de negocio explicando la utilidad de los datos proyectados.

2. Datos

2.1 Contexto

Durante muchos años, en Argentina no se han elaborado ni publicado estadísticas relacionadas al sector aéreo de ningún tipo (pasajeros, movimientos, etc.). Sin embargo, existieron documentos que publicaban algunos organismos en forma esporádica.

Por ejemplo, hasta el año 2008, Aeropuertos Argentina 2000 publicó en su sitio archivos con planillas con información estadística de cada uno de sus aeropuertos, pero luego se discontinuó y se eliminaron los documentos publicados. Por su parte, el Organismo Regulador del Sistema Nacional de Aeropuertos (ORSNA) comenzó en 2014 a publicar información estadística con datos desde el 2001 en formato de informes anuales, pero dado que su actividad se centra en los aeropuertos, existía un doble conteo de pasajeros haciendo que la información brindada sea poco representativa de la actividad en su conjunto. Por último, el Instituto Nacional de Estadística y Censos (INDEC) divulgaba, en forma trimestral, algunos datos de pasajeros (con información de Migraciones), movimientos y carga transportada, pero se discontinuó la actualización de la información en octubre 2015.

La base de datos Sistema Integrado de Aviación Civil (SIAC) es, actualmente, la fuente primaria de información del sector aerocomercial. En dicha base, la carga de datos es manual por parte de operadores de EANA y es administrado por la ANAC. En dichos documentos se cuenta con información de vuelos, movimientos y pasajeros abierta por mes de vuelo, aerolínea y ruta (considerando vuelos regulares y no regulares, sin considerar vuelos Cargo).

2.2. Fuentes de datos

En el presente trabajo se utilizan los datos publicados por ANAC de las series históricas de pasajeros comerciales disponibles desde enero 2001 hasta diciembre 2020[6], abiertas por mes de vuelo. Estos datos tienen una actualización mensual y son publicados por ANAC en su página web. Dado que los datos se obtienen de archivos separados, en primera instancia se procedió a unificarlos para poder contar con la información en una única fuente. Las redes LSTM Univariadas y el modelo TBATS tomarán como datos de entrenamiento y testeo únicamente estas series.

Asimismo, el presente trabajo también explora modelos multivariados a partir del uso de las redes LSTM que permiten realizar proyecciones con más de una variable. En estos modelos se consideran además las siguientes variables:

- Serie histórica de cantidad de vuelos a nivel OD publicada por ANAC[7].
- Serie histórica de Tipo de Cambio de Referencia Comunicación "A" 3500 (Mayorista) publicada por el Banco Central de la República Argentina (BCRA).

[6] Administración Nacional de Aviación Civil (2021). *Tabla 20: Pasajeros Comerciales Internacionales. Tablas de Movimientos y Pasajeros 2001-2018 y Tablas de Movimientos y Pasajeros 2019-2022*. Estadísticas del mercado aerocomercial. Fuente: https://datos.anac.gob.ar/estadisticas/

[7] Administración Nacional de Aviación Civil (2021). *Tabla 22: Vuelos Comerciales Internacionales. Tablas de Movimientos y Pasajeros 2001-2018 y Tablas de Movimientos y Pasajeros 2019-2022.* Estadísticas del mercado aerocomercial. Fuente: https://datos.anac.gob.ar/estadisticas/

- Serie histórica del Estimador Mensual de Actividad Económica (EMAE) publicada por el INDEC[8]

Con la incorporación de esta información se busca entender si el modelo logra mejorar su proyección considerando información relevante de contexto económico. Argentina es un país que atraviesa considerables fluctuaciones económicas que afectan el poder adquisitivo de la población y, por ende, sus decisiones de consumo. Esto se ve reflejado en el comportamiento de estas series, dado que antes fluctuaciones en el tipo de cambio, por ejemplo, los pasajeros pueden alterar sus destinos turísticos (y no tanto aquellos que son viajes de trabajo, concentrados en rutas corporativas).

A partir de esto, los datos con los que se trabajarán los modelos multivariados LSTM son los siguientes:

- Mes-Año
- Cantidad de Pasajeros por ruta
- Cantidad de Vuelos por ruta
- Tipo de Cambio Mayorista BCRA
- Índice Serie Original Estimador Mensual de Actividad Económica

Como universo de análisis, se consideran 20 rutas regionales que concentran más del 50% de los pasajeros regionales-internacionales (en ANAC, la categoría se define como Rutas Internacionales). Asimismo, se busca contemplar dentro del análisis rutas que presenten diferencias en la composición de pasajeros (turístico, corporativo, étnico) como también de estacionalidad, a de modo de poder evaluar la capacidad predictiva de los modelos.

A continuación, se detallan las rutas consideradas.

Tabla 1. Rutas a nivel aeropuerto, origen y destino

#	Ruta Roundtrip
1	Ezeiza <> Santiago de Chile
2	Aeroparque <> Santiago de Chile
3	Ezeiza <> San Pablo
4	Aeroparque <> San Pablo
5	Ezeiza <> Rio de Janeiro
6	Aeroparque <> Rio de Janeiro
7	Ezeiza <> Lima
8	Aeroparque <> Montevideo
9	Ezeiza <> Montevideo
10	Ezeiza <> Santa Cruz de la Sierra
11	Aeroparque <> Santa Cruz de la Sierra
12	Ezeiza <> Asunción
13	Aeroparque <> Asunción
14	Aeroparque <> Punta del Este
15	Ezeiza <> Florianópolis
16	Aeroparque <> Florianópolis
17	Ezeiza <> Salvador de Bahía
18	Aeroparque <> Salvador de Bahía
19	Ezeiza <> Porto Alegre
20	Aeroparque <> Porto Alegre

[8] Instituto Nacional de Estadísticas y Censos (2021). *Estimador mensual de actividad económica (EMAE).* Fuente: https://www.indec.gob.ar/indec/web/Nivel4-Tema-3-9-48

A partir de la Resolución 183/2018 de la ANAC, el 3 de mayo del 2018 comenzó la desregionalización del Aeroparque Jorge Newbery (Aeroparque), trasladando a partir de esa fecha, el 50% de la operación aérea de rutas regionales al Aeropuerto Ministro Pistarini de Ezeiza (Ezeiza). Esta medida tenía como objetivo implementar un proceso gradual de reordenamiento de las operaciones del Aeroparque Jorge Newbery, de acuerdo con las necesidades y características de la creciente operatoria en ese momento y del tráfico proyectado en el marco de la Revolución de los Aviones, que buscaba duplicar la cantidad de personas que viajan en avión[9].

Como consecuencia, Aeroparque pasó de 222 frecuencias semanales de vuelos regionales a 112 frecuencias semanales: el 54% de Aerolíneas Argentinas/Austral, 31% de LATAM, 10% de Gol y 5% Amazonas Paraguay. En la primera etapa, los destinos que siguieron operando desde Aeroparque fueron Curitiba, San Pablo, Santiago de Chile y Asunción.

La segunda etapa se llevó a cabo el 1° de abril de 2019, trasladando el 50% restante. De esta forma, Aeroparque solo quedó operando vuelos de cabotaje y desde/hacia la República Oriental de Uruguay.

A partir de esta resolución, muchos vuelos en conexión desde/hacia el interior del país quedaron desconectados, o empeoraron su servicio, dados los elevados tiempos de transporte entre Aeroparque y Ezeiza. Si bien se reforzaron frecuencias de vuelos de cabotaje en Ezeiza, el nivel de servicio general para estas conexiones se vio afectado.

Por otro lado, a partir del cambio de gobierno a finales de 2019, se dio un nuevo giro a esta dinámica. A principios de febrero 2020, mediante la Resolución 40/2020, se aprobó la regionalización de Aeroparque permitiendo nuevamente a las aerolíneas volver a ofrecer vuelos desde y hacia países limítrofes en Aeroparque a partir del 11 de mayo 2020[10][11]. Esta decisión se enmarcaba en los nuevos objetivos del gobierno entrante de lograr un país más federal, más integrado, buscando dar un mejor servicio al turista ya que más del 60% de los turistas extranjeros eran de países limítrofes. Esta medida apuntaba a beneficiarlos y generar más ingresos por turismo receptivo. Con la pandemia COVID-19 y las obras de refacción y remodelación en Aeroparque, esta decisión se vería efectiva recién a mediados de marzo 2021.

Considerando todo lo expuesto hasta el momento, se puede apreciar que la operación de vuelos regionales ha sufrido diversos cambios entre 2018 y 2020, por lo que en el presente trabajo se decide proyectar a nivel ciudad y no ruta. Esto implica que las rutas que operaron tanto desde Aeroparque como Ezeiza en algún momento serán tomadas en conjunto para evitar quiebres abruptos en la operación y la información de cantidad de vuelos y pasajeros. Como ejemplo, si antes considerábamos Santiago de Chile como:

Ezeiza <> Santiago de Chile

Aeroparque <> Santiago de Chile

[9] Administración Nacional de Aviación Civil) (2018). Desregionalización de Aeroparque: se traspasa el 50% de los vuelos a Ezeiza. Recuperado de: https://www.anac.gov.ar/anac/web/index.php/1/1779/noticias-y-novedades/desregionalizacin-de-aeroparque-se-traspasa-el-50-de-los-vuelos-a-ezeiza

[10] Administración Nacional de Aviación Civil (2020). Aeroparque volverá a operar vuelos regionales. Recuperado de: https://www.anac.gov.ar/anac/web/index.php/1/2049/noticias-y-novedades/-ltstrong-gtaeroparque-volver-a-operar-vuelos-regionales-ltstrong-gt

[11] Sitio oficial del Estado Argentino (2020). *Vuelven los vuelos regionales a Aeroparque.* Recuperado de: https://www.argentina.gob.ar/noticias/vuelven-los-vuelos-regionales-aeroparque

Ahora la consideraremos como:

Buenos Aires <> Santiago de Chile

Esta lógica aplicará a todas las rutas que tengan operación en ambos aeropuertos, mientras que para aquellas que solo operen de alguno de ellos, solo cambiarán el nombre a "Aeroparque o Ezeiza" a "Buenos Aires".

Tabla 2. Rutas a nivel ciudad, origen y destino

Ruta Origen-Destino
Buenos Aires <> Asunción
Buenos Aires <> Florianópolis
Buenos Aires <> Lima
Buenos Aires <> Montevideo
Buenos Aires <> Porto Alegre
Buenos Aires <> Punta del Este
Buenos Aires <> Rio de Janeiro
Buenos Aires <> Salvador de Bahía
Buenos Aires <> Santa Cruz de la Sierra
Buenos Aires <> Santiago de Chile
Buenos Aires <> San Pablo

Por último, dado que en la industria aeronáutica es una práctica común nombrar a los orígenes y destinos de acuerdo con el Código de aeropuertos/ciudad de la Asociación Internacional de Transporte Aéreo (IATA, por sus siglas en inglés)[12], renombraremos las rutas seleccionadas en base a este criterio, quedando como resultante 11 rutas en el análisis a nivel ciudad.

Tabla 3. Rutas a nivel ciudad renombradas con el Código de Aeropuertos/Ciudad de IATA

Ruta Origen-Destino	Código IATA
Buenos Aires <> Asunción	BUE<>ASU
Buenos Aires <> Florianópolis	BUE<>FLN
Buenos Aires <> Lima	BUE<>LIM
Buenos Aires <> Montevideo	BUE<>MVD
Buenos Aires <> Porto Alegre	BUE<>POA
Buenos Aires <> Punta del Este	BUE<>PDP
Buenos Aires <> Rio de Janeiro	BUE<>RIO
Buenos Aires <> Salvador de Bahía	BUE<>SSA
Buenos Aires <> Santa Cruz de la Sierra	BUE<>SRZ
Buenos Aires <> Santiago de Chile	BUE<>SCL
Buenos Aires <> San Pablo	BUE<>SAO

2.3. Análisis exploratorio de datos

Como se mencionó anteriormente, podemos describir los valores de una serie de tiempo en base a sus componentes: tendencia, estacionalidad, ciclo y aleatoriedad. A continuación, se exploran algunas de esas componentes para las series involucradas en el presente estudio, con el fin de poder entender su comportamiento para incorporarlo a los modelos de predicción.

[12] International Air Transport Association (IATA) (2022). *IATA Airline and Location Codes.* Recuperado de: https://www.iata.org/en/services/codes/

Antes, dividimos el grupo de rutas seleccionadas en dos categorías: rutas corporativas y rutas turísticas, quedando dicha clasificación del siguiente modo:

Tabla 4. Rutas a nivel ciudad abiertas por categorías

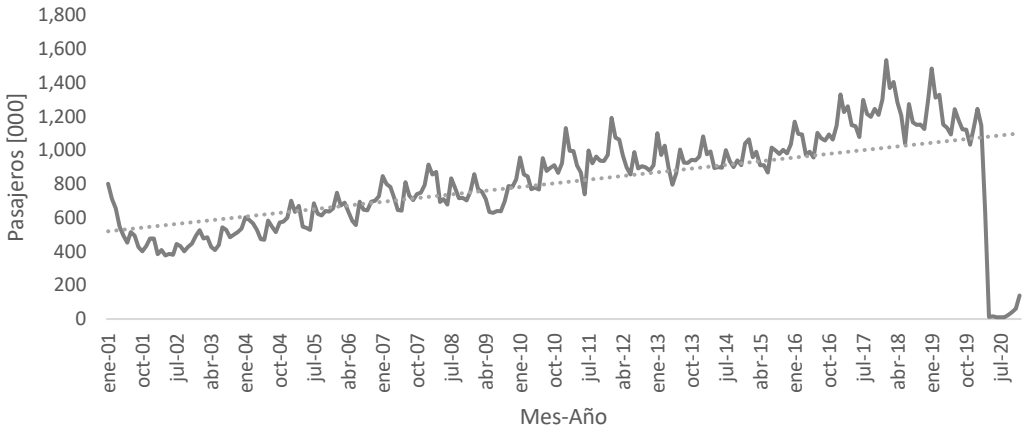
Rutas Corporativas	Rutas Turísticas
BUE<>ASU	BUE<>FLN
BUE<>LIM	BUE<>PDP
BUE<>MVD	BUE<>RIO
BUE<>POA	BUE<>SSA
BUE<>SRZ	
BUE<>SCL	
BUE<>SAO	

Vale la pena aclarar que esta clasificación se realiza en base a conocimientos de la industria aerocomercial y que es en términos generales (predominancia en el *mix* de pasajeros). Existen algunas rutas que, dependiendo del mes, pueden cambiar su composición de pasajeros corporativos/turísticos.

Tendencia

Como primer ejercicio analizamos los datos de pasajeros comerciales regional-internacional desde 2001 hasta 2020, en frecuencia mensual. Una primera aproximación de la tendencia se puede realizar con una estimación lineal. Observando los datos se evidencia un crecimiento continuo a lo largo de los años, a excepción del período de pandemia, tal como se ve en el gráfico a continuación:

Figura 1. Evolución de pasajeros comerciales Regional-Internacional 2001-2020 abierto por mes



Fuente: Elaboración propia en base a datos de ANAC

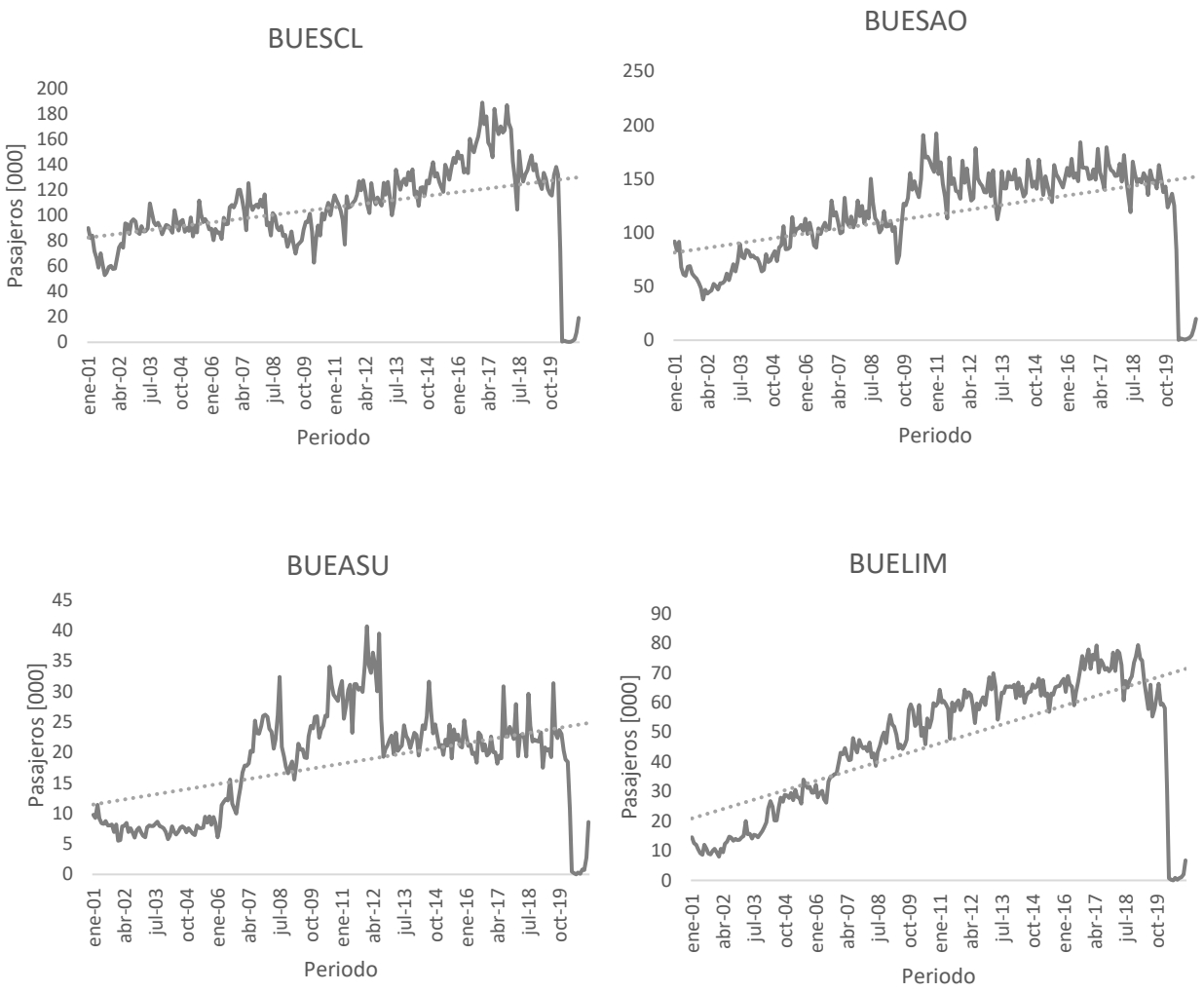
Este crecimiento se puede explicar por diversos factores: desarrollo de la industria aeronáutica local/regional debido a medidas gubernamentales, tipo de cambio apreciado que favorecía el turismo en el exterior y ciclos económicos que impulsaron el tráfico corporativo.

La caída drástica hacia el final del período corresponde a la crisis del COVID y las restricciones de vuelo que fueron impuestas en ese momento, que llevaron a que el número de pasajeros se redujera al punto tal de ser prácticamente nulo a principios/mediados de 2020. Este quiebre tan abrupto en la serie representa todo un desafío para los distintos métodos que buscan predecir la demanda en este período. En particular, un *output* interesante que se deriva de esta tesis es

justamente evaluar el desempeño de los distintos métodos ante la presencia de importantes quiebres estructurales.

A partir de esto, procedemos a analizar la tendencia en las rutas corporativas. A primera vista podemos observar que rutas como BUESCL, BUESAO, BUELIM y BUEASU presentan una marcada tendencia creciente:

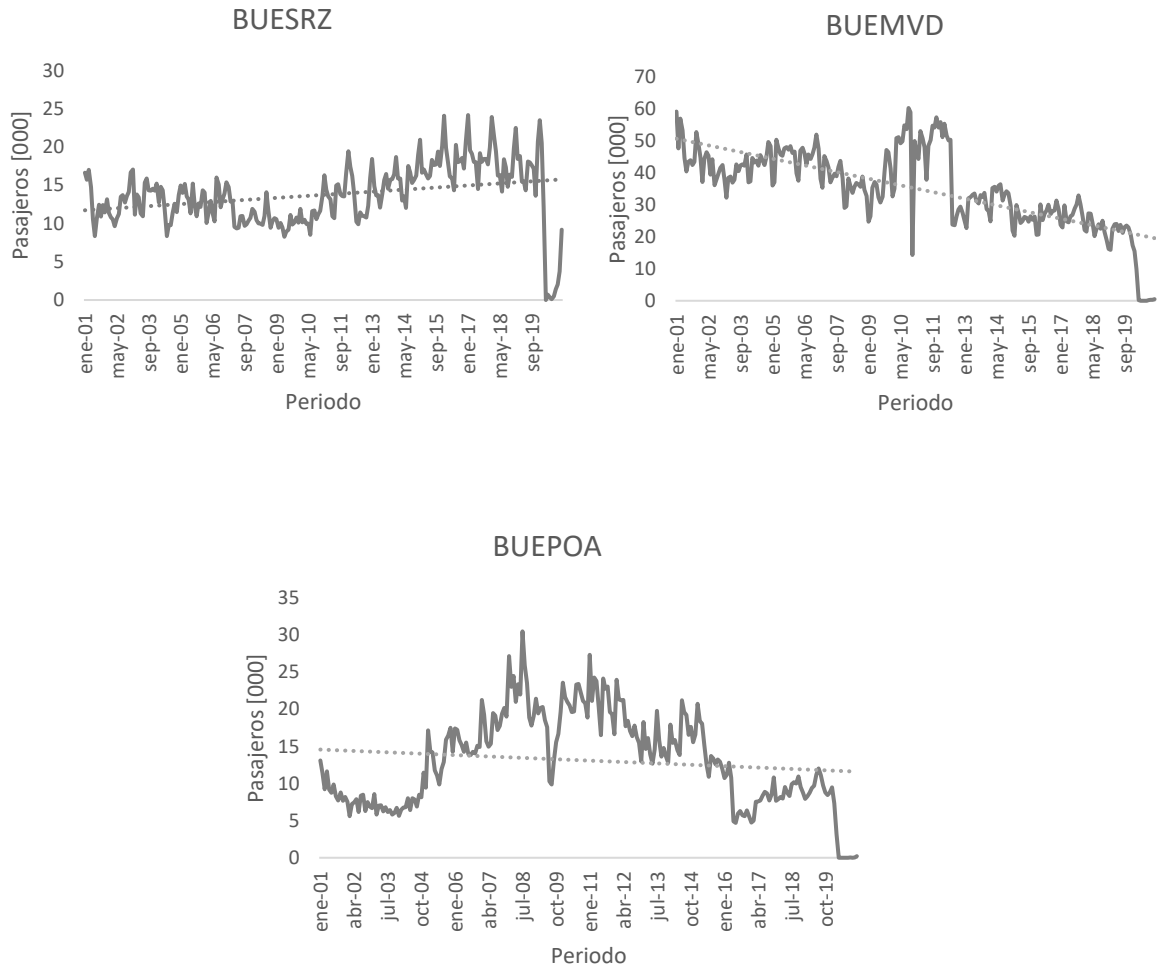
Figura 2. Evolución de pasajeros comerciales rutas corporativas 2001-2020 abierto por mes: BUESCL, BUESAO, BUEASU y BUELIM



Fuente: Elaboración propia en base a datos de ANAC

Mientras, que BUESRZ presenta una tendencia positiva, pero mucho menos marcada que las rutas mencionadas hasta ahora. Por otro lado, BUEMVD y BUEPOA presentan una tendencia negativa a lo largo de 2001-2020.

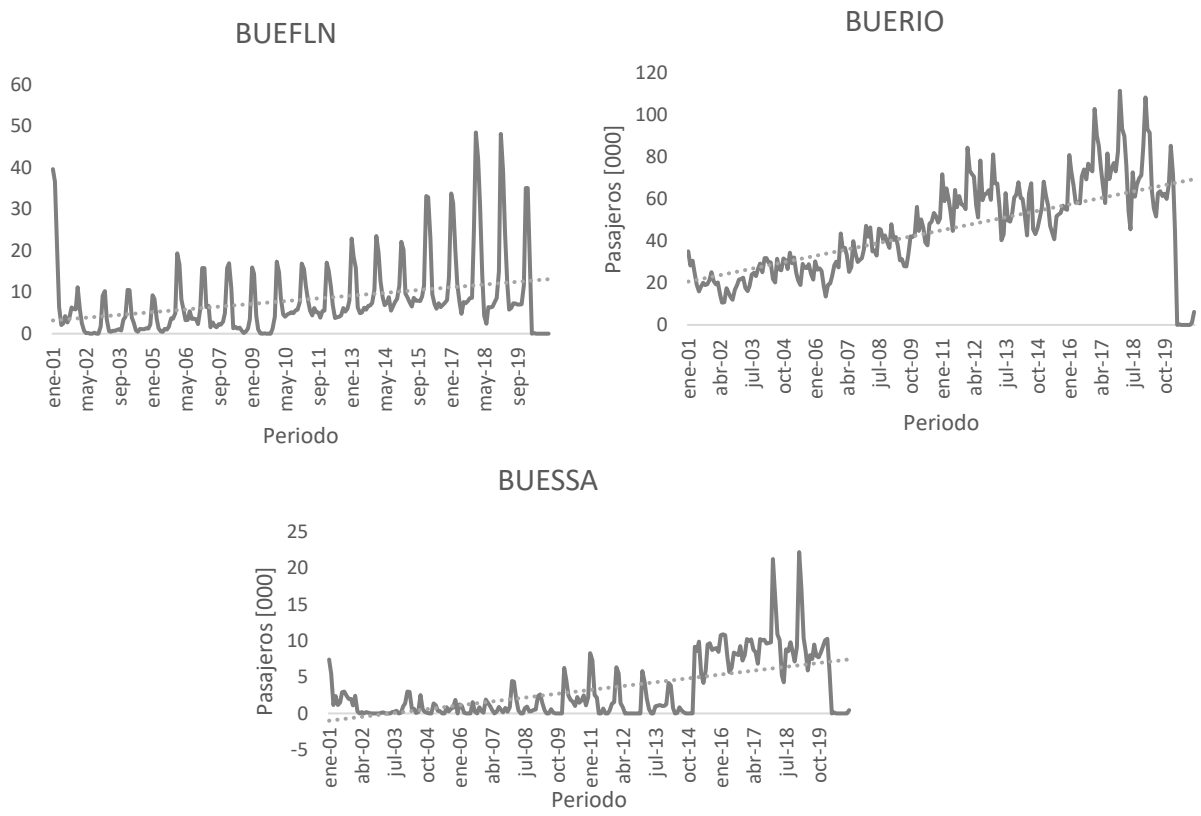
Figura 3. Evolución de pasajeros comerciales rutas corporativas 2001-2020 abierto por mes: BUESRZ, BUEMVD y BUEPOA



Fuente: Elaboración propia en base a datos de ANAC

Por otro lado, analizando las rutas turísticas vemos que BUERIO, BUEFLN y BUESSA presentan tendencia creciente.

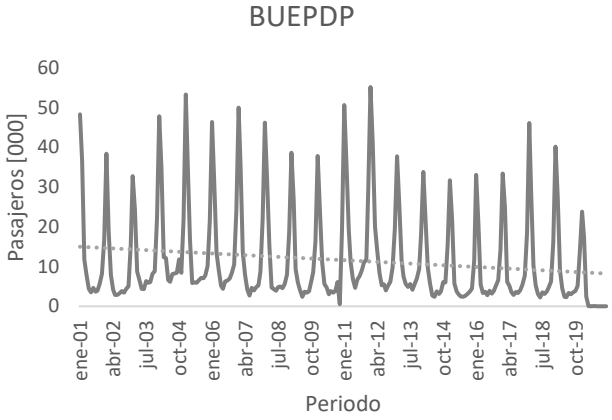
Figura 4. Evolución de pasajeros comerciales rutas turísticas 2001-2020: BUEFLN, BUERIO y BUESSA



Fuente: Elaboración propia en base a datos de ANAC

Por el contrario, BUEPDP presenta tendencia decreciente.

Figura 5. Evolución de pasajeros comerciales rutas turísticas 2001-2020: BUEPDP



Fuente: Elaboración propia en base a datos de ANAC

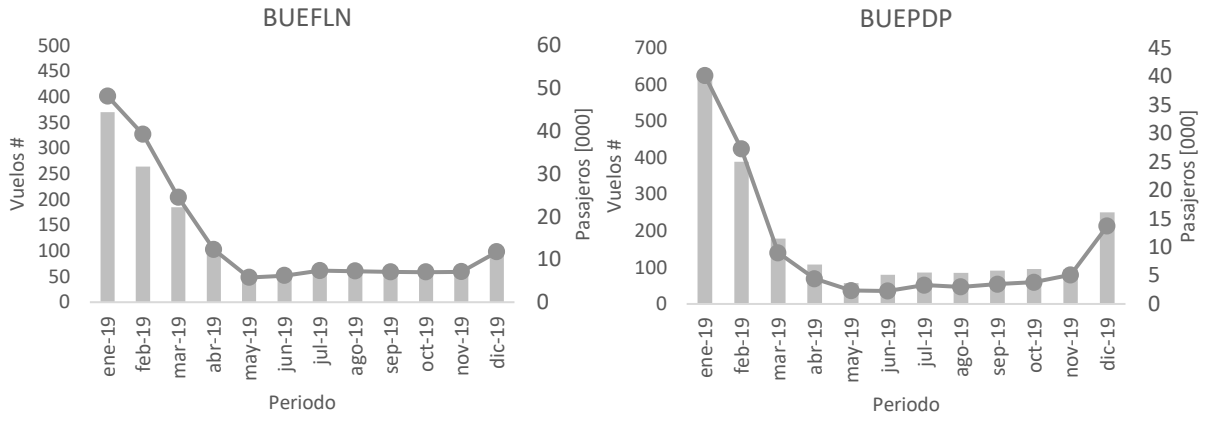
La pendiente de la tendencia en un período dado depende del incremento (o decremento) del número de pasajeros a lo largo de los meses. Se debe, generalmente, a cuestiones exógenas

tales como la entrada de una nueva aerolínea, aumento de frecuencias y oferta de vuelos, apertura de nuevas rutas desde y hacia este aeropuerto, campañas turísticas, promociones particulares que pueden impulsar fuertemente un período dado, etc. Adicionalmente, el crecimiento demográfico y la situación económica, tanto nacional como propia de la región, también están vinculados a este comportamiento. Por ejemplo, un determinado valor del tipo de cambio hace que los argentinos hagan más turismo interno que viajes al exterior, o viceversa, y de igual manera para los visitantes extranjeros.

Estacionalidad

La cantidad de vuelos y pasajeros suelen estar fuertemente correlacionados con la época del año. Este comportamiento se evidencia más marcadamente en las rutas turísticas, donde hay un crecimiento pronunciado en meses de temporada. Si observamos, por ejemplo, BUEPDP y BUEFLN en el año 2019, vemos que en los meses de verano (diciembre, enero y febrero) la cantidad de vuelos y pasajeros crece considerablemente:

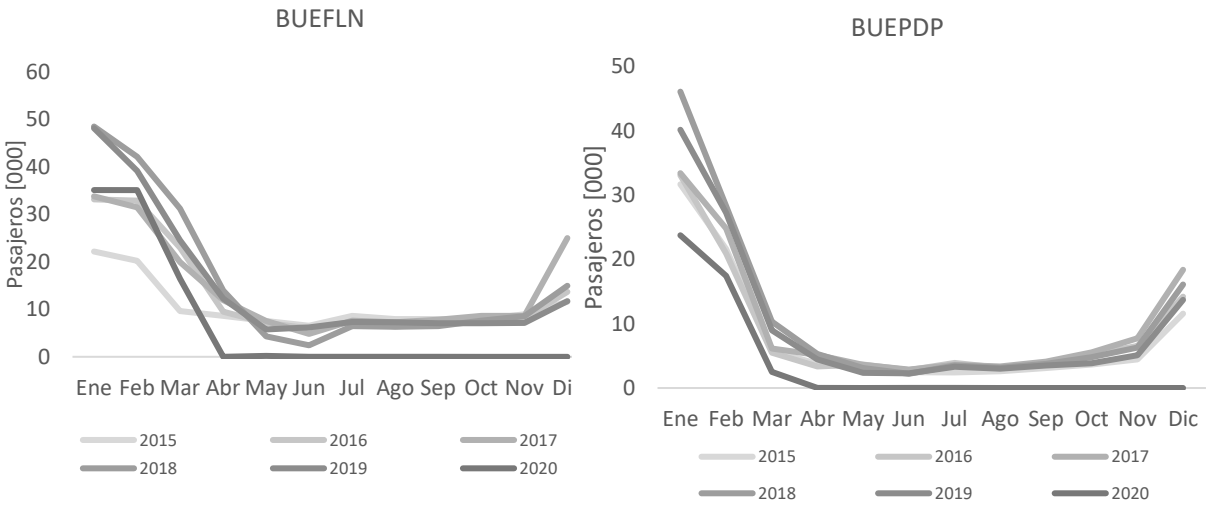
Figura 6. Cantidad de pasajeros y vuelos 2019 rutas turísticas abierto por mes: BUEFLN y BUEPDP



Fuente: Elaboración propia en base a datos de ANAC

Adicionalmente, si observamos el comportamiento de pasajeros en los últimos años podemos ver el comportamiento estacional:

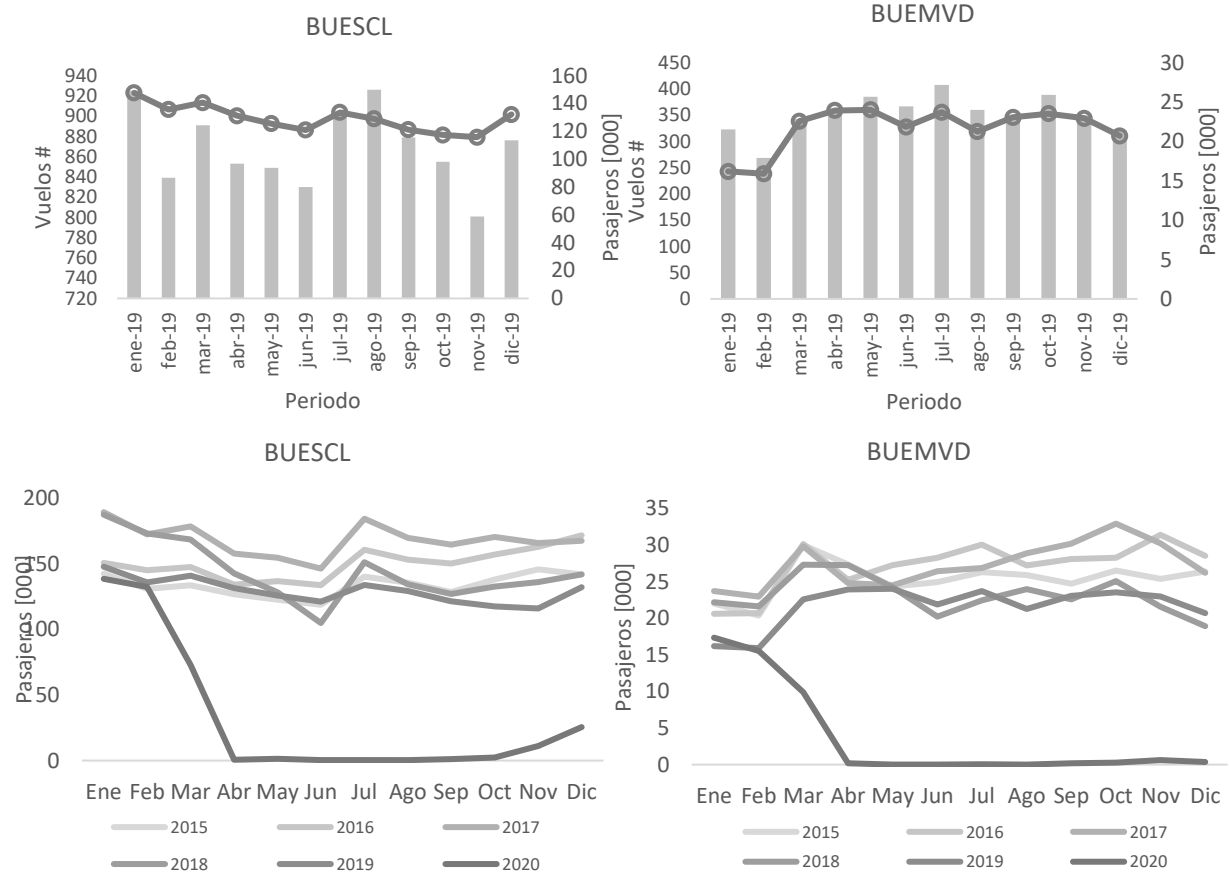
Figura 7. Evolución de pasajeros de rutas turísticas 2015-2020 abiertos por mes



Fuente: Elaboración propia en base a datos de ANAC

Por otro lado, observando las rutas corporativas no tenemos una estacionalidad tan marcada como si se observa en las rutas turísticas.

Figura 8. Cantidad de pasajeros y vuelos 2019 rutas turísticas abierto por mes: BUESCL y BUEMVD



Fuente: Elaboración propia en base a datos de ANAC

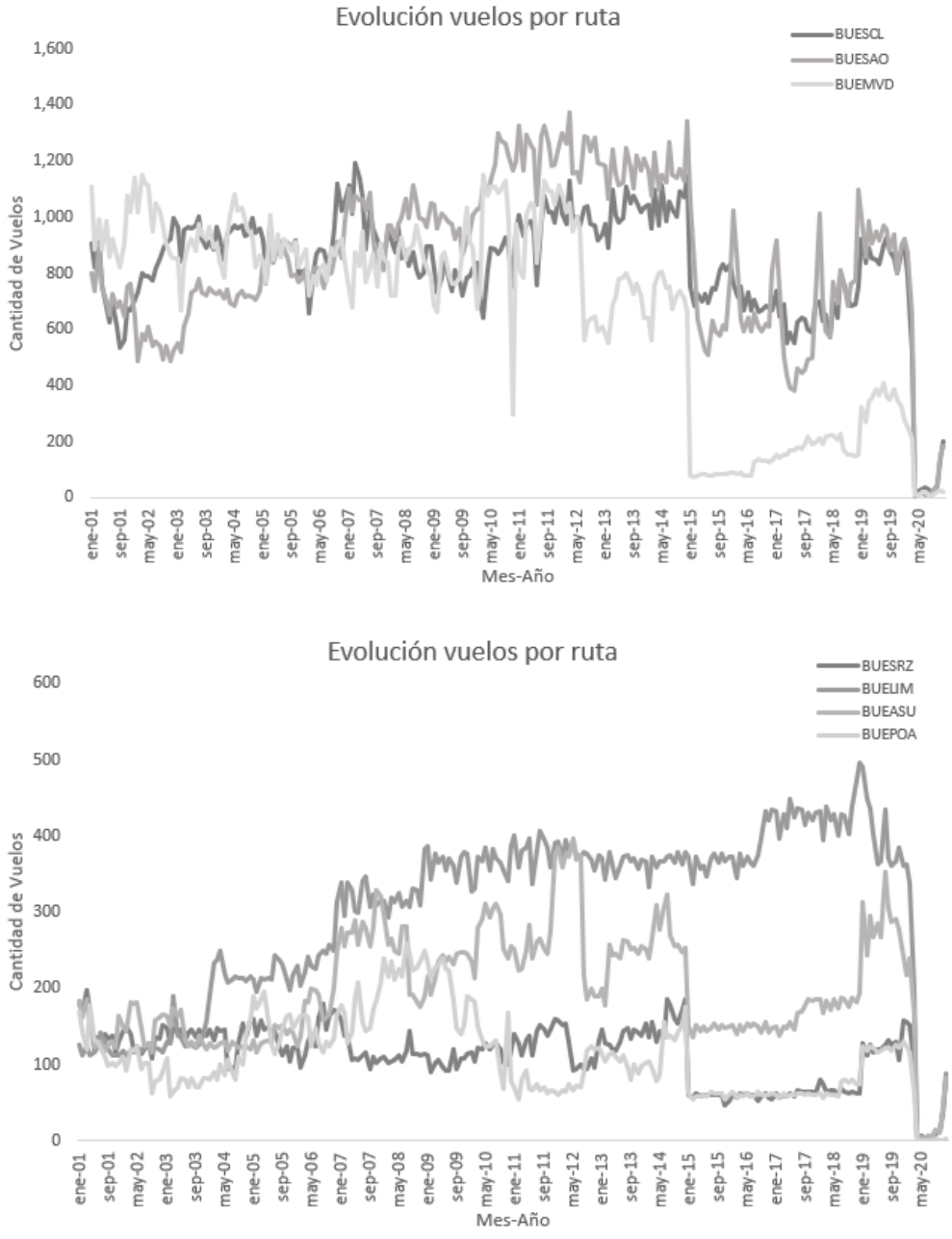
Como consecuencia, al trabajar con series temporales de pasajeros comerciales la longitud de los periodos va a ser de 12 meses.

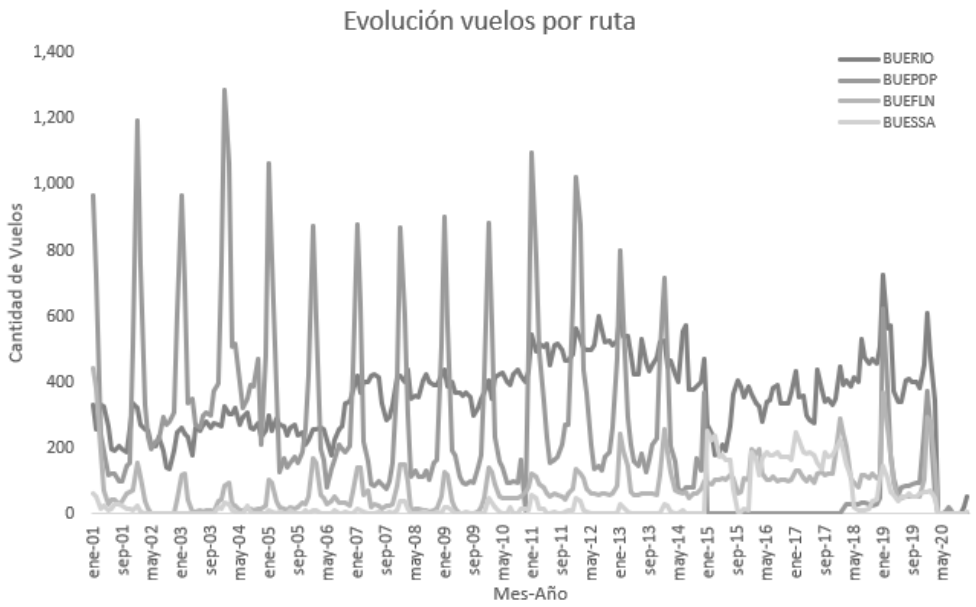
Otras series de datos utilizados en los modelos

Para las redes LSTM Multivariadas se utilizan como input, además de la serie de pasajeros, otras series temporales que se cree pueden explicar también a esa serie. Tal es el caso de cantidad de vuelos por ruta OD, la serie de Tipo de Cambio Mayorista y el EMAE.

Analizando la evolución de la cantidad de vuelos para las rutas que se analizan en el presente análisis, vemos una evolución positiva en la mayoría de las rutas hasta el 2019.

Figura 9. Cantidad de vuelos 2001-2020 rutas corporativas y turísticas abierto por mes

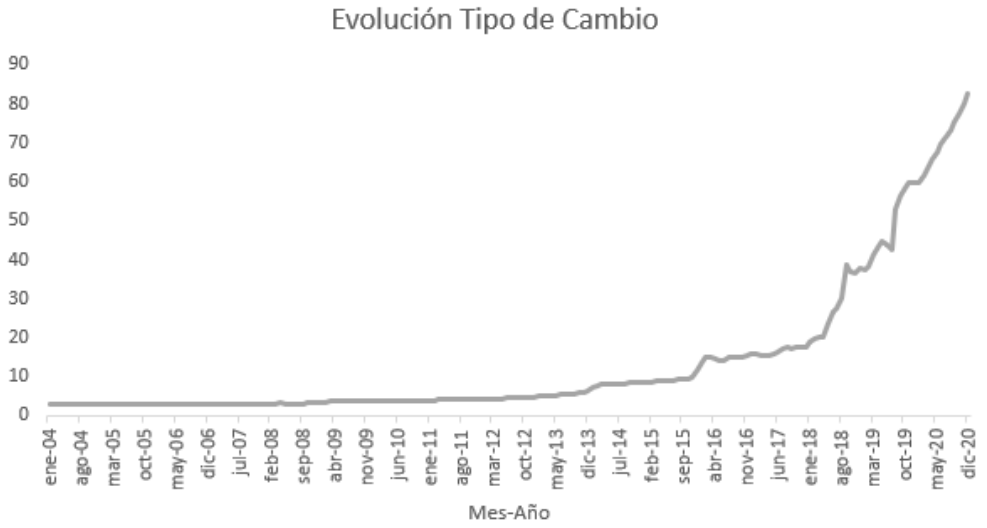


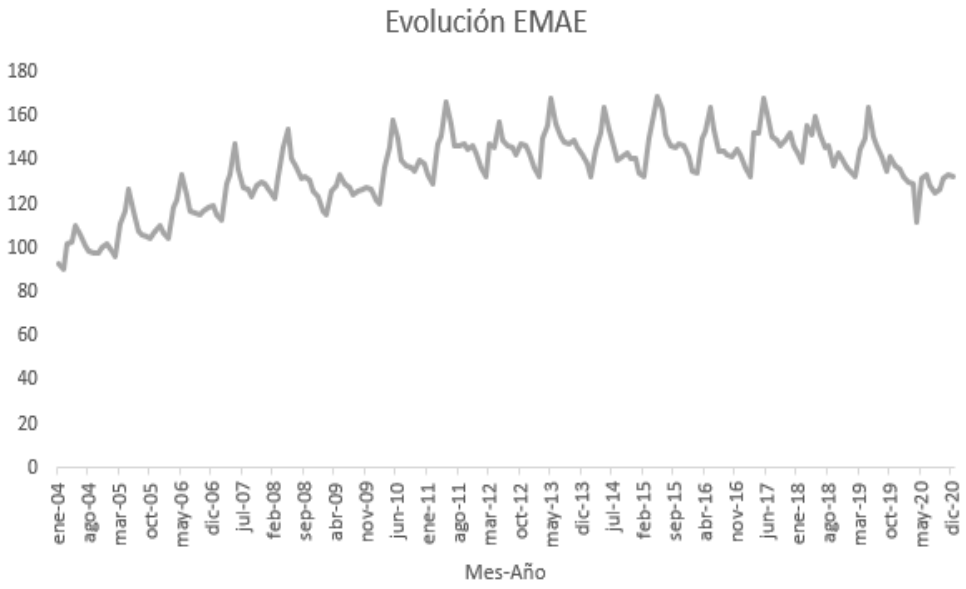


Fuente: Elaboración propia en base a datos de ANAC

Por otro lado, si observamos la serie del Tipo de Cambio bilateral vemos un crecimiento más marcado a partir del 2018, si bien venía creciendo años anteriores, desde ese año se aceleró el crecimiento y el ritmo de devaluación. Por su parte, el EMAE, que exhibe un claro comportamiento estacional, muestra una caída hacia el año 2020 producto del impacto económico de la Pandemia COVID-19.

Figura 10. Evolución del Tipo de Cambio Mayorista y del Estimador Mensual de Actividad Económica.





Fuente: Elaboración propia en base a datos del INDEC y BCRA

3. Metodología

3.1. Pronósticos de series de tiempo

En la mayoría de las empresas la previsión resulta vital para lograr una planificación eficiente y eficaz de los recursos. Por esta razón los pronósticos de ciertas variables resultan de gran importancia en la toma de decisiones.

Existen diversas variables que pueden ser relevantes en este aspecto y predecirlas puede requerir diferentes grados de dificultad. La predictibilidad de un evento o cantidad puede depender de diversos factores, tales como:

- El nivel de entendimiento de los factores que contribuyen
- La disponibilidad de los datos
- Si los pronósticos pueden afectar lo que estamos tratando de proyectar

En el presente trabajo se busca pronosticar o proyectar pasajeros a nivel origen destino para ciertas rutas regionales operadas en Argentina. Entendemos que se cumple la segunda condición respecto a la disponibilidad de datos, pero se tiene un conocimiento limitado sobre los factores que afectan a la demanda de pasajeros y el pronóstico de pasajeros puede incluso influir en la demanda futura.

Los buenos pronósticos capturan los patrones y relaciones genuinas que existen en los datos históricos, pero no replican eventos pasados que no volverán a ocurrir. Se asume erróneamente que los pronósticos no son buenos en un entorno cambiante. Todos los entornos están cambiando y un buen modelo de pronóstico debería capturar la forma en que están cambiando las cosas. Lo que los modelos normalmente asumen es que la forma en que el entorno está cambiando continuará en el futuro. Es decir, un entorno muy volátil seguirá siendo muy volátil. Un modelo de pronóstico tiene como objetivo capturar la forma en que se mueven las cosas, no solo dónde están (Hyndman y Athanasopoulos, 2021)[13].

Los métodos de pronóstico pueden ser simples, como usar las observaciones más recientes como pronóstico, o muy complejos, como el uso de redes neuronales. El pronóstico de una serie temporal consiste en extender los valores históricos hacia el futuro. Este ejercicio está definido por el periodo, es decir, el nivel de agregación (horas, días, meses, etc.) y el horizonte, es decir, la cantidad de periodos a proyectar.

En general, se conocen dos tipos de pronósticos:

- ***Predicción a un paso***: utiliza la información del pasado y pronostica el valor siguiente.
- ***Predicción recursiva de largo plazo***: requerirá utilizar valores calculados por el propio modelo para la estimación siguiente.

En el presente estudio, se busca hacer una proyección a un horizonte de 12 meses y, por lo tanto, nos encontramos en el segundo caso. El problema que puede existir en tal situación usando una predicción recursiva es que, justamente porque utiliza otras estimaciones, el error

[13] Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice*. 3rd edition, OTexts: Melbourne, Australia.

se puede propagar rápidamente y el intervalo de confianza será mayor cuanto más lejos se encuentre el período a proyectar con respecto al último dato histórico disponible.

3.2. Modelos de series temporales

Existen diferentes métodos o técnicas de análisis de series de tiempo. Estos métodos consideran si los datos históricos se encuentran correlacionados, la tendencia subyacente y la estacionalidad de la serie. Entre las técnicas estadísticas y métodos clásicos para proyecciones de series de tiempo más relevantes se encuentran:

- ***Modelos de regresión:*** se pronostica la serie temporal *Y* a partir de una relación lineal o no lineal con otras series de tiempo comprendidas en una matriz *X*.
- ***Métodos de pronósticos y suavizamiento exponencial:*** son promedios ponderados de observaciones pasadas, con los pesos decayendo exponencialmente a medida que las observaciones son más lejanas en el tiempo.
- ***Modelos SARIMA:*** son modelos univariados que buscan capturar la dinámica propia de la serie que se quiere pronosticar modelándola a partir de sus propios rezagos y rezagos de shocks pasados. Son una extensión de los modelos ARIMA (modelos autorregresivos integrados de medias móviles) que permiten modelar a su vez, la estacionalidad cambiante en el tiempo, de ahí se deriva el nombre SARIMA por estacionalidad ("S" = *seasonality*).
- ***Métodos TBATS – BATS:*** permiten modelar series de tiempo con múltiples estacionalidades. Se los denomina BATS como un anacronismo de las características claves del modelo: transformación Box y Cox, errores ARMA, componentes de tendencia y estacionalidad. El modelo TBATS se refiere a una transformación trigonométrica del modelo BATS para conseguir un enfoque más flexible aplicable a modelos con frecuencia estacional no entera.

Por otro lado, en los últimos años han surgido técnicas alternativas de aprendizaje automatizado (*machine learning*) que permitieron encontrar buenos resultados. Las redes neuronales *(Neural Networks*, NN por sus siglas en inglés*)* son modelos que aprenden automáticamente, detectando patrones complejos entre los datos y en base a ello, predicen valores futuros de la serie. El aprendizaje automático se realiza a través del tiempo, incorporando nuevos datos y valores de salida que se vuelven a entrenar y calibrar los pesos relativos de los factores del modelo.

En el presente trabajo se utilizarán los modelos TBATS y redes neuronales para proyectar los pasajeros de rutas regionales a un año, 12 meses en avance y se contrastarán los resultados obtenidos para entender si las redes neuronales logran captar mejor que métodos mas tradicionales como TBATS, el contexto de pandemia en el año 2020.

3.3. Redes neuronales

Las redes neuronales son modelos de aprendizaje automático que buscan simular virtualmente el comportamiento del cerebro humano. Tienen su origen en el Perceptrón, creado por Frank Rosenblatt en el año 1958.

Un perceptrón simple se compone de varias entradas binarias $x_1, x_2, ..., x_d$ produciendo una sola salida *y*. La salida es calculada a partir de la relación lineal entre las entradas y los "pesos" $w_1, w_2, ..., w_d$, que son un número real que expresa la importancia de la respectiva entrada con la

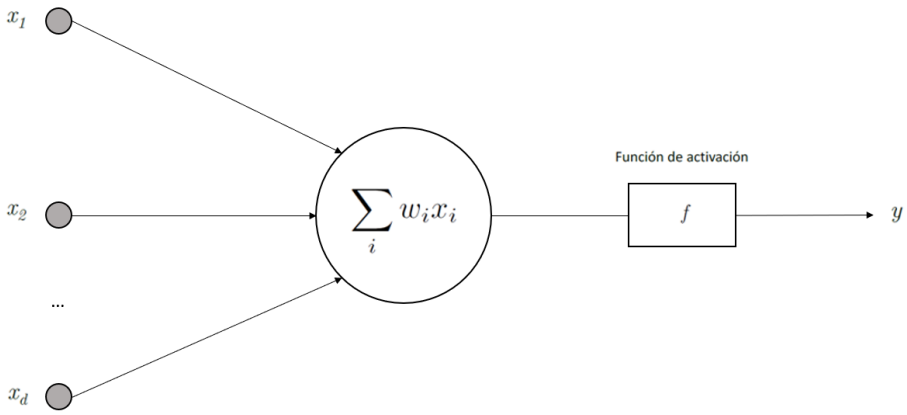
salida. La salida de la neurona será 1 o 0 si la suma de la multiplicación de pesos por entradas es mayor o menor a un determinado umbral[14]. Podemos definir la función de activación como:

Ecuación 1. Función de activación de una red neuronal

$$y = f\left(\sum_{i=1}^{d} w_i x_i - b\right) = f\left(\sum_{i=1}^{d} w_i x_i\right) = f(\langle w, x \rangle)$$

Donde *b* es el sesgo o *bias* y *d* es el número de entradas.

Figura 11. Perceptrón simple



Fuente: Elaboración propia en base a Rosenblatt (1958)

Podemos pensar a las redes neuronales como redes de neuronas organizadas en capas, donde en el caso del perceptrón simple, no existen capas ocultas. Este modelo es equivalente a una regresión lineal y el pronóstico de la variable se obtiene como una combinación lineal de los *inputs*. Los pesos óptimos se obtienen a partir de un algoritmo de aprendizaje que minimiza la función de costo (por ejemplo: el Error Cuadrático Medio, *MSE* por sus siglas en inglés, o la Raíz del Error Cuadrático Medio, *RMSE* por sus siglas en inglés).

A medida que se agregan capas ocultas la red neuronal se convierte en no lineal y se conoce como red neuronal multicapa (*Multilayer Perceptron, MLP* por sus siglas en inglés). En estas redes, cada capa oculta recibe como entrada los *outputs* de las capas anteriores y las entradas de cada neurona son combinaciones lineales ponderadas. Esos resultados son luego modificados por una función no lineal antes de pasar a la siguiente capa. Esto tiende a reducir el efecto de los valores de entrada extremos, volviendo más robusta a la red a los valores atípicos.

Estas redes tienen la capacidad de aprender por sí solas a medida que ingresa nueva información y el modelo se va adaptando, variando el peso de las conexiones.

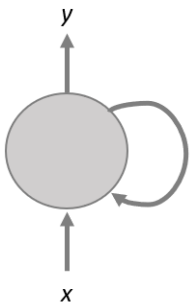
[14] Na8 (2018). *Breve historia de las Redes Neuronales Artificiales.* Aprende Machine Learning Blog. Recuperado de: https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/

3.4. Redes neuronales recurrentes (RNN)

Las redes neuronales recurrentes o *Recurrent Neural Networks* (*RNN* por sus siglas en inglés), son una clase de redes para analizar datos de series temporales permitiendo tratar la dimensión de "tiempo", es decir, reconociendo que las observaciones no son independientes entre sí.

A diferencia de las redes neuronales, cuya función de activación solo actúa en una dirección, hacia adelante, las RNN incluyen conexiones "hacia atrás". Una RNN simple, con una única neurona se puede expresar como una red que recibe una entrada, produciendo una salida y enviando esa salida a sí misma (Figura 12).

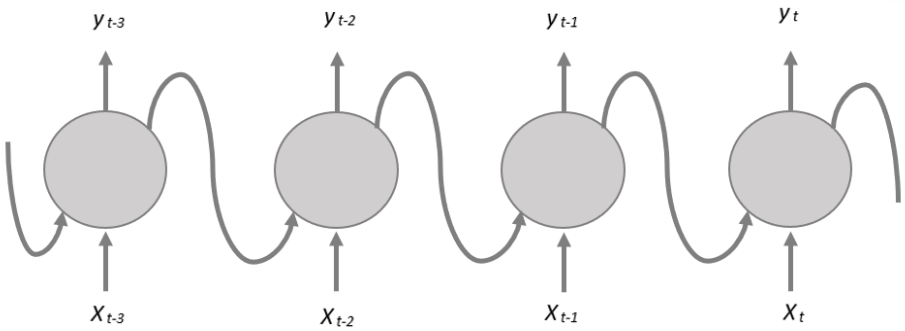
Figura 12. Red Neuronal Recurrente simple



Fuente: Elaboración propia en base a Python Deep Learning: Introducción práctica con Keras y TensorFlow 2 (2020)

En cada instante de tiempo (*timestep*), esta neurona recurrente recibe la entrada x_t de la capa anterior, así como su propia salida del instante de tiempo anterior y_{t-1} para generar su salida y_t. Una nueva capa de neuronas recurrentes se puede implementar de tal manera que, en cada instante de tiempo, cada neurona recibe dos entradas: la entrada correspondiente de la capa anterior y a su vez la salida del instante anterior de la misma capa.

Figura 13. Red Neuronal Recurrente con múltiples capas



Fuente: Elaboración propia en base a Python Deep Learning: Introducción práctica con Keras y TensorFlow 2 (2020)

De este modo, cada neurona recurrente tiene dos conjuntos de parámetros:

- Uno que lo aplica a la entrada de datos que recibe de la capa anterior: *x*

- Otro conjunto que lo aplica a la entrada de datos correspondiente al vector salida del instante anterior: y_{t-1}

Pudiéndose expresar con la siguiente formula:

Ecuación 2. Función de activación red neuronal recurrente

$$y_t = f\,(W_{xt} + U_{yt} - 1 + b)$$

Donde $x = (x_1, \ldots, x_T)$ representa la secuencia de entrada proveniente de la capa anterior, *W* los pesos de la matriz y *b* el sesgo (*bias*) vistos ya en las anteriores capas. Las RNN extienden esta función con una conexión recurrente en el tiempo donde U es la matriz de pesos que opera sobre el estado de la red en el instante de tiempo anterior (y_{t-1}). En el entrenamiento de la red neuronal recurrente, a través de *backpropagation,* se actualizan los pesos de esta matriz que minimizan la función de pérdida o costo.

Dado que la salida de una neurona recurrente en un instante de tiempo determinado es una función de entradas de los instantes de tiempo anteriores, se podría decir que una neurona recurrente tiene en cierta forma memoria. La parte de una red neuronal que preserva un estado a través del tiempo se denomina *memory cell* (o simplemente *cell*).

Esta "memoria interna" es lo que hace de este tipo de redes muy adecuadas para problemas de aprendizaje automático que involucran datos secuenciales. Gracias a esta, las RNN pueden recordar información relevante sobre la entrada que recibieron, lo que les permite ser más precisas en la predicción de lo que vendrá después manteniendo información de contexto a diferencia de los otros tipos de redes, que no pueden recordar acerca de lo que ha sucedido en el pasado, excepto lo reflejado en su entrenamiento a través de sus pesos.

En las redes neuronales convencionales se hace *forward-propagation* para obtener el resultado de aplicar el modelo y verificar si este resultado es correcto o incorrecto para obtener la pérdida. Después se hace *backward-propagation (o backpropagation),* que implica ir hacia atrás a través de la red neuronal para encontrar las derivadas parciales del error con respecto a los pesos de las neuronas. Esas derivadas son utilizadas por el algoritmo *Gradient Descent* para minimizar iterativamente una función dada, ajustando los pesos hacia arriba o hacia abajo, dependiendo de cómo se disminuye la función de pérdida o *loss function*.

Con *backpropagation* se ajustan los pesos del modelo mientras se entrena las redes convencionales, pero en las RNN se ajustan con *Backpropagation Through Time* (BPTT), que incluye la dimensión tiempo dentro del proceso.

Al realizar el proceso de BPTT, se requiere a nivel matemático incluir la conceptualización de desenrollar, ya que la función de pérdida de un determinado instante de tiempo depende del instante (*timestep*) anterior. Dentro de BPTT, el error es propagado hacia atrás desde el último hasta el primer instante de tiempo, mientras se desenrollan todos los instantes de tiempo. Esto permite calcular la función de pérdida para cada instante de tiempo, lo que permite actualizar los pesos.

Long-Short Term Memory (LSTM)

Los modelos *Long-Short Term Memory (LSTM)* son una extensión de las redes neuronales recurrentes, que básicamente amplían su memoria para aprender de experiencias importantes que han pasado hace mucho tiempo. Las LSTM permiten a las RNN recordar sus entradas

durante un largo período de tiempo, debido a que los modelos LSTM contienen su información en la memoria, que puede considerarse similar a la memoria de una computadora, en el sentido que una neurona de una LSTM puede leer, escribir y borrar información de su memoria.

Esta memoria se puede ver como una celda bloqueada, donde bloqueada significa que la célula decide si almacenar o eliminar información dentro (abriendo la puerta o no para almacenar), en función de la importancia que asigna a la información que está recibiendo. La asignación de importancia se decide a través de los pesos, que también se aprenden mediante el algoritmo. Esto se puede traducir como que el modelo LSTM aprende con el tiempo qué información es importante y cuál no.

En una neurona LSTM existen tres puertas a estas "celdas" de información: puerta de entrada (*input gate*), puerta de olvidar (*forget gate*) y puerta de salida (*output gate*). Estas puertas determinan si se permite o no una nueva entrada, se elimina la información porque no es importante o se deja que afecte a la salida en el paso de tiempo actual.

Las puertas en un modelo LSTM son análogas a una forma sigmoide, lo que significa que van de 0 a 1. El hecho de que sean análogas a una función de activación sigmoide permite incorporarlas al proceso de *backpropagation*. Los problemas de desvanecimiento de los gradientes o *Vanishing Gradients* se resuelven a través del modelo LSTM porque mantiene los gradientes lo suficientemente empinados y, por lo tanto, el entrenamiento es relativamente corto y la precisión alta.

Como se mencionó anteriormente, en el presente trabajo se utilizarán los modelos TBATS y redes neuronales para proyectar los pasajeros de rutas regionales a un año, 12 meses en avance y se contrastarán los resultados obtenidos para entender si las redes neuronales logran captar mejor que otros métodos el contexto reciente. Se espera que los modelos LSTM puedan proyectar mejor que otros modelos dada su capacidad de memoria y de incorporar información reciente y pasada en las estimaciones.

3.5. Implementación de modelos LSTM y TBATS

En esta sección se abordará la implementación, creación y entrenamiento de los modelos LSTM y TBATS. Se detallan las transformaciones realizadas al *dataset* utilizado en el presente trabajo para poder adaptarlo al *input* que necesitan los modelos. Asimismo, se brinda detalles del código y las librerías utilizadas para la transformación de los datos y la implementación de los modelos.

3.5.1. Modelo LSTM Univariado

Esta red neuronal LSTM, al ser univariada, será entrenada únicamente con información histórica de la serie temporal que queremos predecir.

Transformación de los datos

Para la implementación de la red neuronal LSTM simple, utilizamos la información histórica de pasajeros desde 2001-2020 y la librería *Keras de Tensorflow*. Trabajamos con el *dataset* de pasajeros de ANAC, expresado en miles de pasajeros, abierto por ruta a nivel aeropuerto (como se explica en la sección 2. Datos).

En primer lugar, agregamos ese *dataset* pasando del nivel aeropuerto a nivel ciudad, unificando aquellas rutas que volaban a ambos aeropuertos de Buenos Aires. Por ejemplo, para el caso de BUE<>SCL, se sumaron los pasajeros de las rutas Aeroparque <> Santiago de Chile y Ezeiza <> Santiago de Chile.

Una vez realizada esa primera transformación, se procedió a separar el *dataset* de entrenamiento y prueba. En este primer modelo, separamos como entrenamiento, los datos desde 2001 a 2018 y como prueba, los datos desde 2019 a 2020.

Dado que consideramos un horizonte temporal largo para entrenar, donde los pasajeros en 2018 son considerablemente más altos que los que existían en 2001, para poder introducir los datos al modelo necesitamos normalizarlos. Para realizar esto utilizamos la clase *MinMaxScaler* de la librería S*klearn,* con la que normalizamos la información a valores entre 0 y 1.

Con las redes LSTM queremos predecir el valor en el *time step* actual usando *n time steps* previos, por lo que necesitamos ajustar los datos de entramiento y testeo. Para ello se crea una función llamada *create_dataset* que se encarga de realizar la división de datos en ambos conjuntos:

```
def create_dataset(X, y, time_steps=1):
Xs, ys = [], []
for i in range(len(X) - time_steps):
v = X.iloc[i:(i + time_steps)].values
Xs.append(v)
ys.append(y.iloc[i + time_steps])
return np.array(Xs), np.array(ys)
```

Dado que trabajamos con series temporales de pasajeros, utilizaremos pasos de tiempo o *time steps* de 12 meses (12 observaciones), por lo que definimos una variable llamada *time steps* = 12. Con esto remodelaremos la información de entrenamiento y testeo para poder utilizarla directamente en el modelo con el siguiente formato: *[muestras, pasos de tiempo, características]*

Una vez definida la función creamos los *datasets* de entrenamiento (*train*) y prueba (*test*), resultando del siguiente modo:

```
X_train: Array (204, 12, 1)
y_train: Array (204,)
X_test: Array (12, 12, 1)
y_test: Array (12,)
```

La forma de entrada será 12 pasos de tiempo con 1 características o *feature*

Creación del modelo

Luego de ajustar los datos, se procede a crear la red LSTM. Se constituye como un modelo secuencial, con 1 capa visible, una capa oculta con 100 bloques LSTM o neuronas y una capa de salida. Se utiliza el método de la ventana, dado que se usan los 12 pasos anteriores para predecir el paso actual. Se utiliza por defecto la función de activación sigmoide para los bloques de memoria LSTM.

Luego se definen la función de perdida y de optimización. Usamos entropía cruzada o *cross entropy loss* como función de pérdida. Para la función de optimización usamos *Adam optimizer.*

La tasa de aprendizaje o *learning rate* fue definida en 0.01. La red LSTM queda definida entonces de la siguiente manera:

```
model = keras.Sequential()

model.add(keras.layers.LSTM(100, input_shape=(X_train.shape[1],
X_train.shape[2])))

model.add(keras.layers.Dense(1))

model.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(0.01))
```

Entrenamiento del modelo

Se puede cuantificar el proceso de aprendizaje como la reducción del resultado de la función de pérdida. En las redes neuronales cada ciclo de corrección de propagación hacia atrás y hacia adelante para reducir la pérdida se denomina época o *epoch*. Por esto, la propagación hacia atrás consiste en determinar las mejores ponderaciones y sesgos de entrada para obtener un resultado más preciso o "minimizar la pérdida"[15].

En esta primera iteración se definieron 100 épocas para entrenar el modelo y un tamaño de lote o *batch size* de 42. Resulta importante destacar que el modelo da distintos resultados cada vez que se ejecuta, porque se inicializa el modelo con diferentes pesos en la función de optimización.

```
history = model.fit(
  X_train, y_train,
  epochs=100,
  batch_size=42,
  verbose=1,
  shuffle=False)
```

Este procedimiento se realizó para las 11 rutas OD a nivel ciudad comprendidas en el presente análisis.

3.5.2. Modelo LSTM Multivariado

Transformación de los datos

Para la implementación de la red neuronal LSTM Múltiple, utilizamos la siguiente información económica y la librería *Keras de Tensorflow*:

- Cantidad de Pasajeros por ruta - ANAC

[15] Frank La Vigne(2019). *¿Cómo aprenden las redes neuronales?*. Microsoft Documentation. Recuperado de: https://docs.microsoft.com/es-es/archive/msdn-magazine/2019/april/artificially-intelligent-how-do-neural-networks-learn#:~:text=Cada%20ciclo%20de%20correcci%C3%B3n%20de,o%20%22minimizar%20la%20p%C3%A9rdida%22.

- Cantidad de Vuelos por ruta - ANAC
- Tipo de Cambio Mayorista - BCRA
- Índice Serie Original Estimador Mensual de Actividad Económica - INDEC

Al igual que con los datos en la red LSTM Univariada, se unificaron los pasajeros y vuelos del d*ataset* pasando de nivel aeropuerto a nivel ciudad. Dado que el EMAE se encuentra publicado desde 2004, se ha decidido descartar los datos de 2001-2003 para este modelo, resultando un *dataset* de 204 observaciones.

Dado que se utiliza un horizonte largo para entrenar y se incorporan nuevas variables al entrenamiento se procedió a normalizar las variables. Para realizar esto utilizamos la clase *MinMaxScaler* de la librería *Sklearn*, llevando la información a valores entre 0 y 1.

Una vez realizada esa primera transformación, se procedió a separar los *datasets* de entrenamiento y prueba. Separamos como entrenamiento, los datos desde 2004 a 2018 y como prueba, los datos desde 2019 a 2020. Luego se remodelan los datos al formato 3 dimensiones que ya mencionamos anteriormente, donde la red necesita como *input*: [muestras, pasos de tiempo, características].

Al igual que con la red LSTM Univariada, preparamos los datos de los conjuntos de entrenamiento y testeo utilizando la función *create_dataset*:

```
def create_dataset(X, y, time_steps=1):
Xs, ys = [], []
for i in range(len(X) - time_steps):
v = X.iloc[i:(i + time_steps)].values
Xs.append(v)
ys.append(y.iloc[i + time_steps])
return np.array(Xs), np.array(ys)
```

Una vez definida la función creamos los datasets, resultando del siguiente modo:

```
X_train: Array (168, 12, 4)
y_train: Array (168,)
X_test: Array (12, 12, 4)
y_test: Array (12,)
```

Creación del modelo

Luego de ajustar los datos, se procede a crear la red LSTM Multivariada. Se constituye como un modelo secuencial, con 1 capa visible, una capa oculta con 100 bloques LSTM o neuronas y una capa de salida. Se utiliza el método de la ventana, dado que se usan los 12 pasos anteriores para predecir el paso actual. Se utiliza por defecto la función de activación sigmoide para los bloques de memoria LSTM.

Luego se definen la función de pérdida y de optimización. Usamos entropía cruzada o *cross entropy loss* como función de perdida. Para la función de optimización usamos *Adam optimizer.*

La tasa de aprendizaje o *learning rate* fue definida en 0.01. La red LSTM multivariada queda definida entonces de la siguiente manera:

```
model = keras.Sequential()

model.add(keras.layers.LSTM(100, input_shape=(X_train.shape[1],
X_train.shape[2])))

model.add(keras.layers.Dense(1))

model.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(0.01))
```

La forma de entrada será 12 paso de tiempo con 4 características o *features.*

Entrenamiento del modelo

En esta primera iteración se definieron 100 épocas para entrenar el modelo y un tamaño de lote o *batch size* de 42. Resulta importante destacar que el modelo da distintos resultados cada vez que se ejecuta, porque se inicializa el modelo con diferentes pesos en la función de optimización.

```
history = model.fit(
   X_train, y_train,
   epochs=100,
   batch_size=42,
   verbose=1,
   shuffle=False)
```

Este procedimiento se realizó para las 11 rutas comprendidas en el presente análisis.

3.5.3. Modelo TBATS

TBATS es un método de pronóstico para modelar datos de series temporales, cuyo objetivo principal es pronosticar series de tiempo con patrones estacionales complejos utilizando suavización exponencial. Este modelo será entrenado únicamente con información de la serie temporal que queremos predecir, es decir, en forma univariada.

Transformación de los datos

Se utiliza el *dataset* de pasajeros de ANAC, en miles de pasajeros, abierto por ruta a nivel aeropuerto (como se explica en la sección 2. Datos) y se unifican para trabajarlas a nivel ciudad. Para la implementación de este modelo, no se requieren adaptaciones de los datos originales. Únicamente se divide el *dataset* en conjunto de entrenamiento y prueba. Considerando data de 2001-2019 para entrenamiento y la data de 2020 para testear:

```
y_to_train: Series (228,0)
y_to_test: Series (12,0)
```

Creación del modelo

Luego se procede a la creación del modelo TBATS. Al tratarse de información mensual, donde cada observación se corresponde a un mes del año, se define la estacionalidad del modelo como mensual y anual:

```
estimator = TBATS (seasonal_periods = (1, 12))
```

Entrenamiento del modelo

Una vez definido el modelo TBATS se procede al entrenamiento, pasándole como *input* la serie de datos de entrenamiento:

```
fitted_model = estimator.fit(y_to_train)
```

Este procedimiento se realizó para las 11 rutas comprendidas en el presente análisis.

4. Resultados

4.1. Proyecciones año 2020

Luego de entrenados los modelos con los datos de entrenamiento, se realizaron las proyecciones para 2020. Para poder comparar los resultados obtenidos entre los diferentes modelos se utilizó el Error Cuadrático Medio (RMSE por sus siglas en ingles) de los datos de *testeo* y las proyecciones realizadas por los modelos.

En la siguiente tabla comparamos los valores del RMSE para la red LSTM Univariada, la Multivariada y el modelo TBATS. De esta primera comparación vemos que la red LSTM Univariada se desempeña mejor que el resto de los modelos, presentando el menor RMSE, aunque aún en valores promedios elevados para rutas como BUESCL, BUESAO, BUELIM y BUERIO.

Tabla 5. Proyecciones 2020: RMSE abierto por ruta

Ruta	RMSE Proyección 2020		
	LSTM Univariada	LSTM Multivariada	TBATS
BUESCL	46.6	108.7	104.1
BUESAO	46.9	99.3	110.6
BUEMVD	5.0	9.5	20.3
BUESRZ	5.9	12.6	13.2
BUELIM	22.7	49.2	50.3
BUEASU	5.1	8.1	17.9
BUERIO	23.9	60.3	52.2
BUEPDP	5.3	11.3	6.5
BUEFLN	6.8	18.0	9.7
BUESSA	3.4	7.1	6.4
BUEPOA	2.0	3.6	8.1

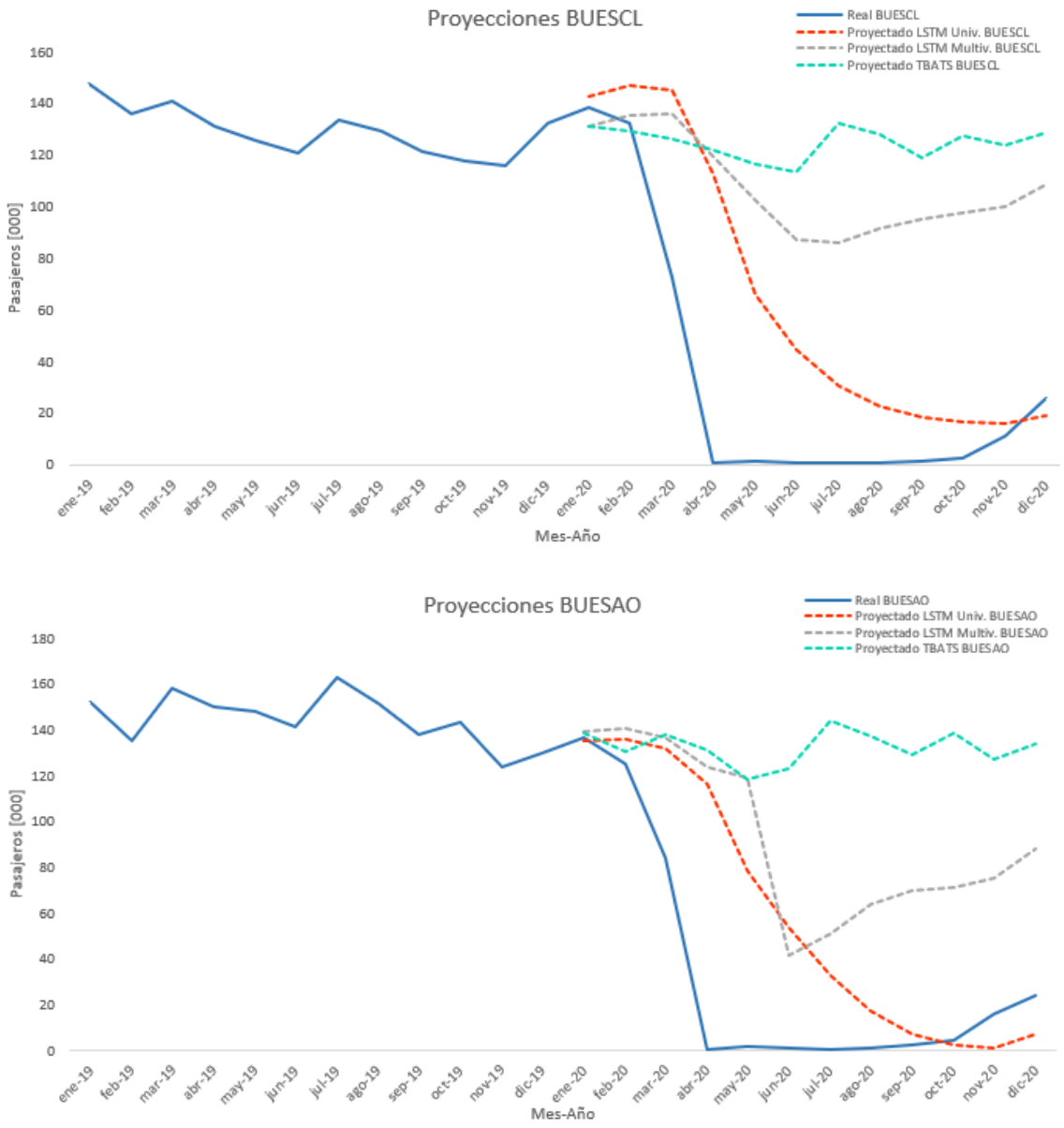
Es de esperar, que los modelos tengan dificultades en predecir el comportamiento del año 2020, dado que se trata de un suceso único por la Pandemia a nivel mundial que, para la industria en cuestión, implicó la cancelación de vuelos y un quiebre de una magnitud histórica en la serie a

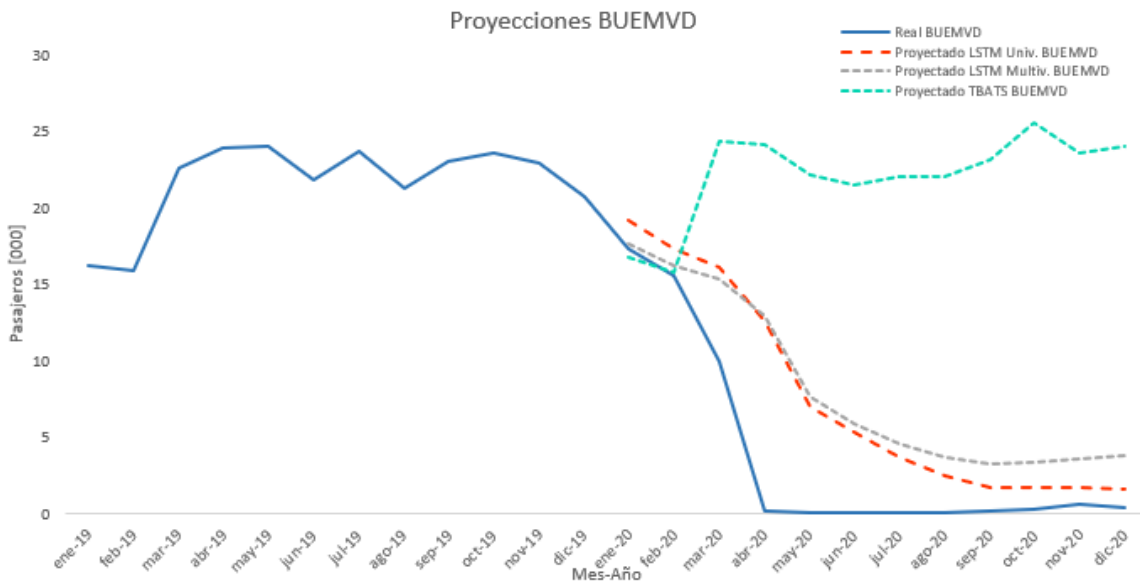
proyectar. A pesar de ello, las redes neuronales LSTM mostraron un mejor desempeño que los modelos tradicionales como TBATS, dado que, al tener memoria, pueden incorporar la historia reciente y generar un pronóstico más cercano a la realidad.

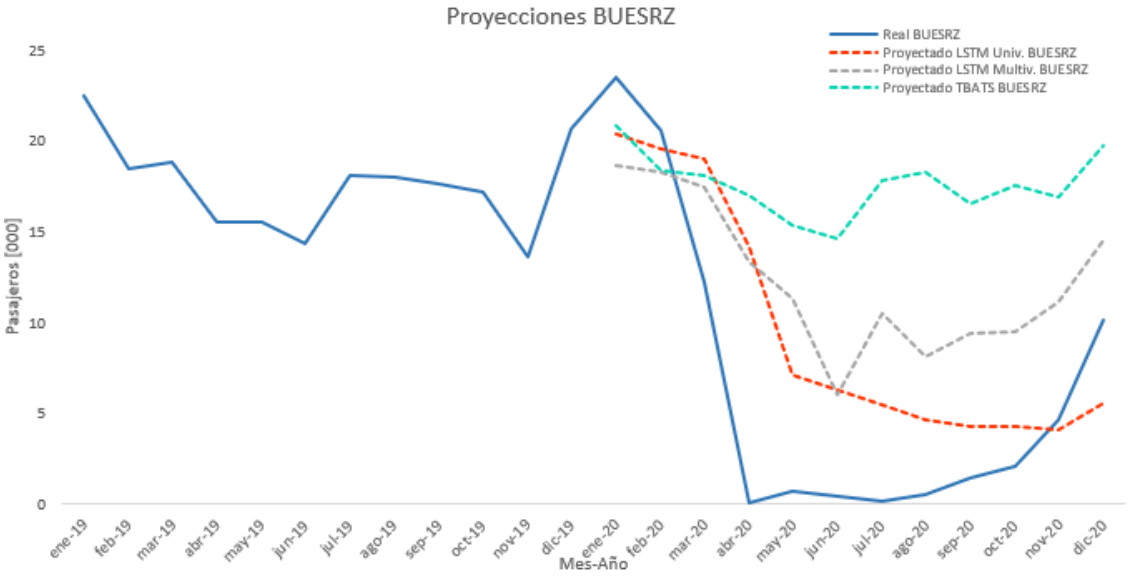
A continuación, se exponen las proyecciones de los tres modelos implementados en el presente trabajo para los pasajeros de 2020 para las once rutas en análisis. Se puede observar, en todos los casos, como las redes LSTM logran capturar, con cierto rezago, la caída abrupta de principios del año 2020 por el COVID-19, y ajustar las proyecciones de los meses siguientes a valores mas bajos, cercanos a la realidad.

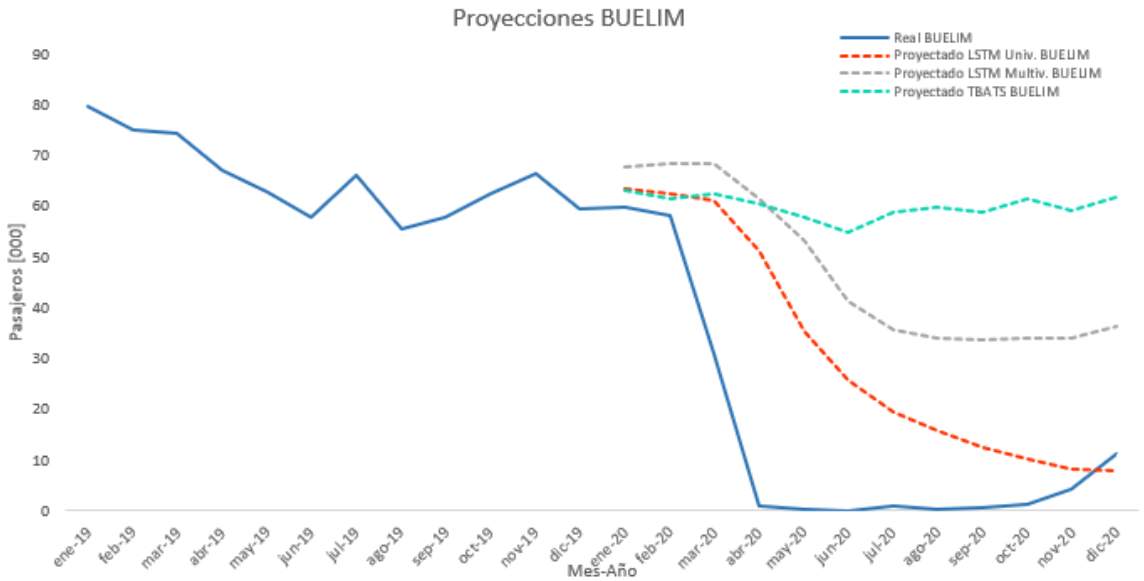
Por su parte, el modelo TBATS, no logra capturar dicha caída abrupta, manteniendo las proyecciones en niveles cercanos a los del 2019.

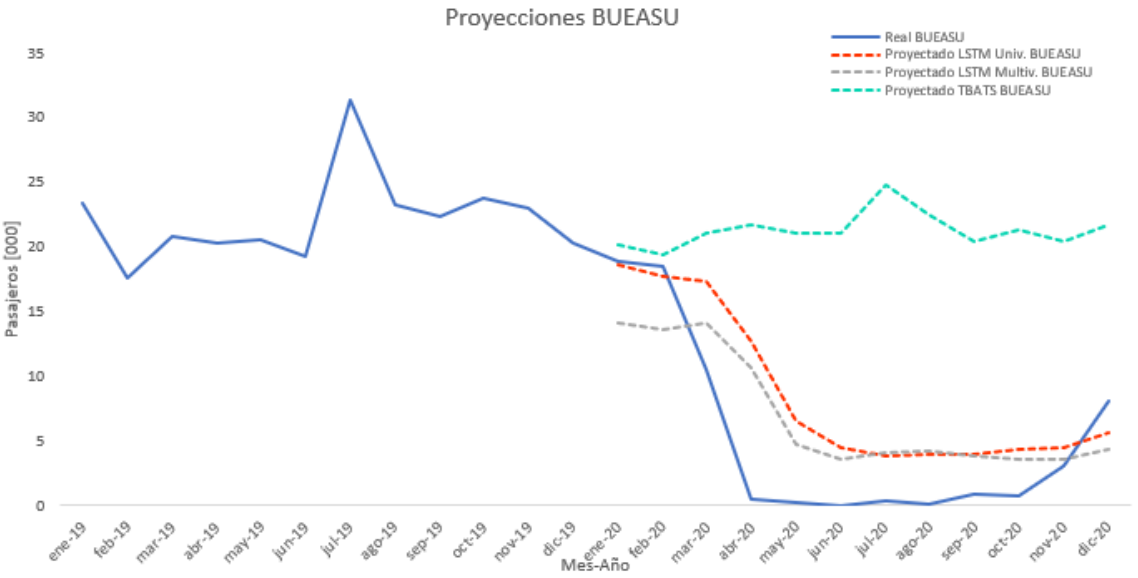
Figura 14. Proyecciones para 2020: Red LSTM Univariada, Multivariada y Modelo TBATS

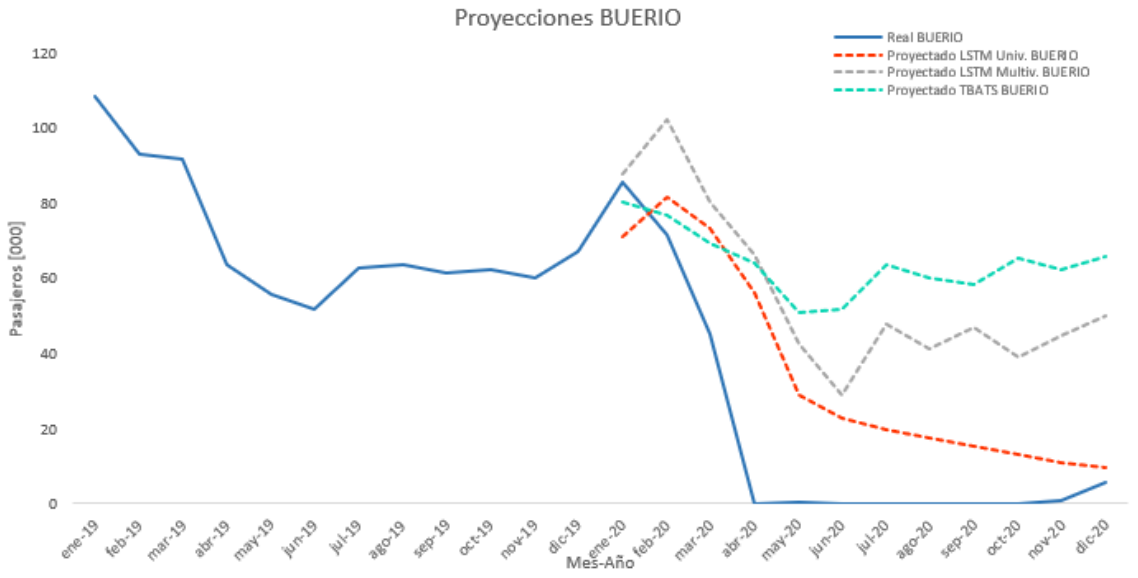


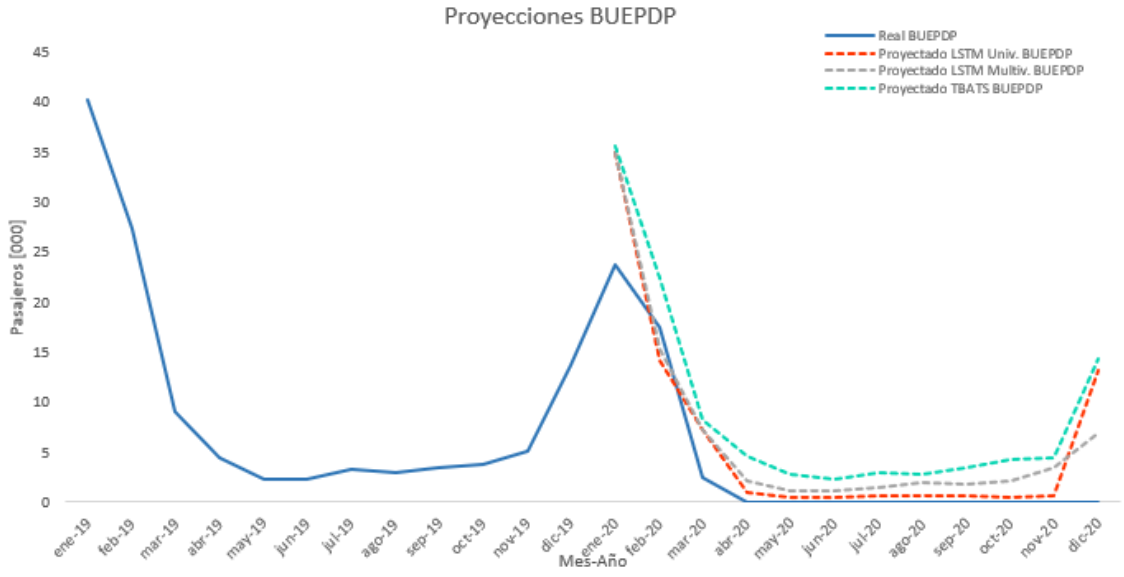

Proyecciones BUEMVD
Real BUEMVD
Proyectado LSTM Univ. BUEMVD
Proyectado LSTM Multiv. BUEMVD
Proyectado TBATS BUEMVD
Pasajeros [000]
Mes-Año

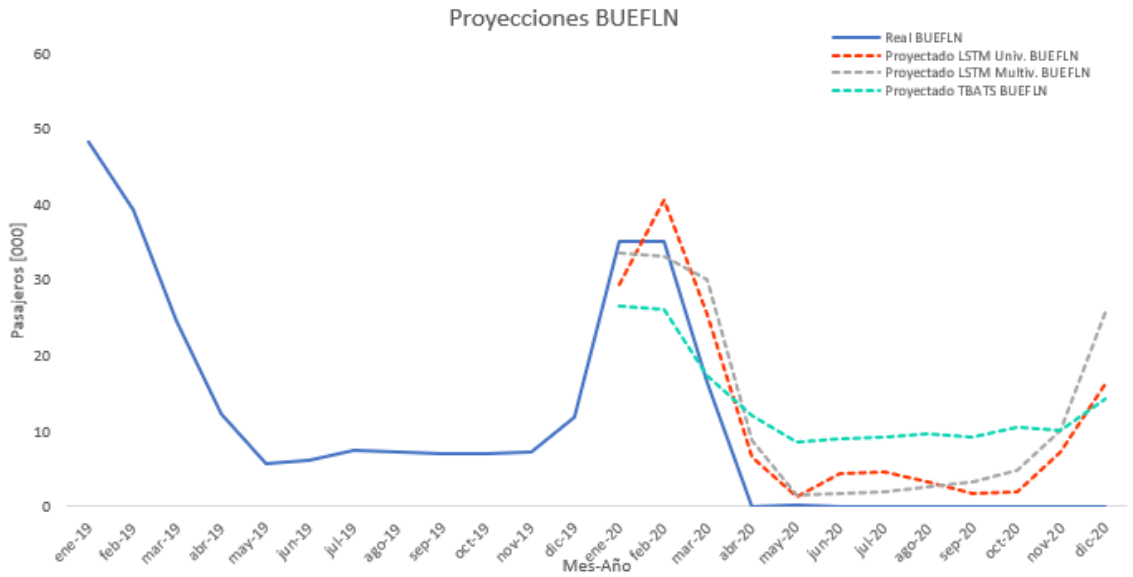

Proyecciones BUESRZ
Real BUESRZ
Proyectado LSTM Univ. BUESRZ
Proyectado LSTM Multiv. BUESRZ
Proyectado TBATS BUESRZ
Pasajeros [000]
Mes-Año

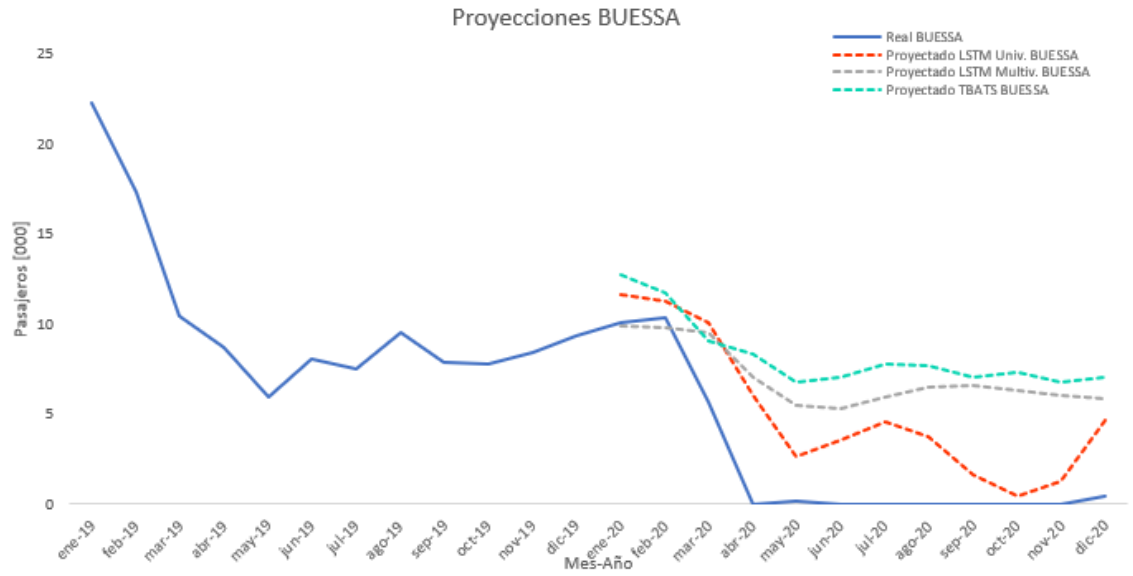

Proyecciones BUELIM
Real BUELIM
Proyectado LSTM Univ. BUELIM
Proyectado LSTM Multiv. BUELIM
Proyectado TBATS BUELIM
Pasajeros [000]
Mes-Año

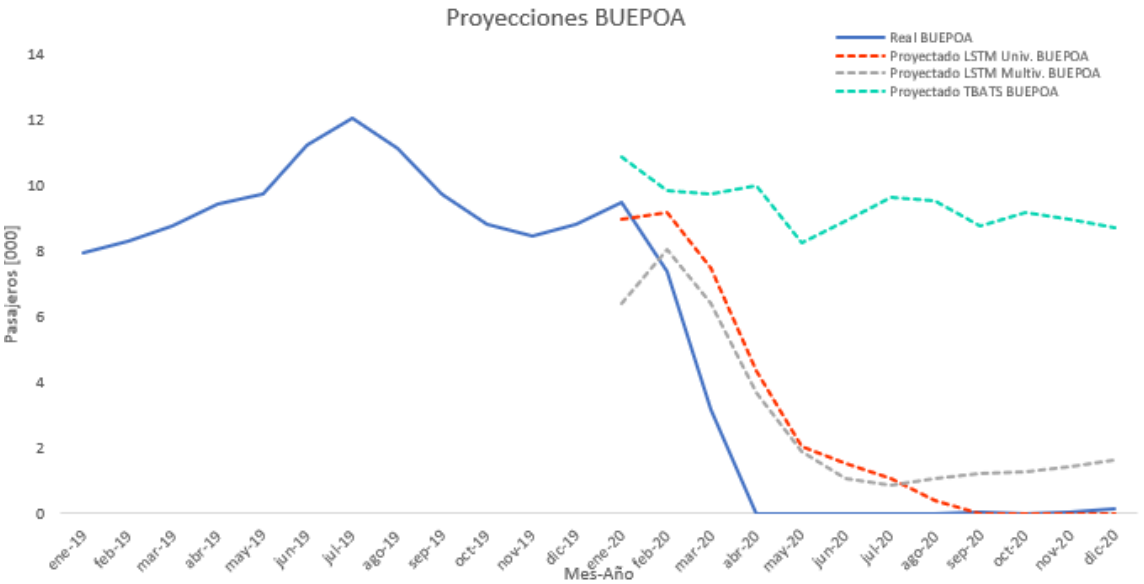

Proyecciones BUEASU
Real BUEASU
Proyectado LSTM Univ. BUEASU
Proyectado LSTM Multiv. BUEASU
Proyectado TBATS BUEASU
Pasajeros [000]
Mes-Año


Proyecciones BUERIO
Real BUERIO
Proyectado LSTM Univ. BUERIO
Proyectado LSTM Multiv. BUERIO
Proyectado TBATS BUERIO
Pasajeros [000]
Mes-Año


Proyecciones BUEPDP
Real BUEPDP
Proyectado LSTM Univ. BUEPDP
Proyectado LSTM Multiv. BUEPDP
Proyectado TBATS BUEPDP
Pasajeros [000]
Mes-Año







Fuente: Elaboración propia en base a datos de ANAC y proyecciones propias

4.2. Actualización de las proyecciones: proyecciones año 2021

Al comienzo del presente trabajo, en el año 2020, solo se contaba con información del año 2020, por lo que se definió como horizonte de predicción el año 2021. Se debería entrenar los modelos con la información hasta el año 2020 y luego realizar las predicciones para el año 2021.

Dado que ahora se cuenta con los datos completos del año 2021, se decidió entrenar los modelos también con dicha información, aprovechando además la vuelta a cierta normalidad en los vuelos y volver a comparar resultados. De este modo, se define como horizonte de predicción al año 2022.

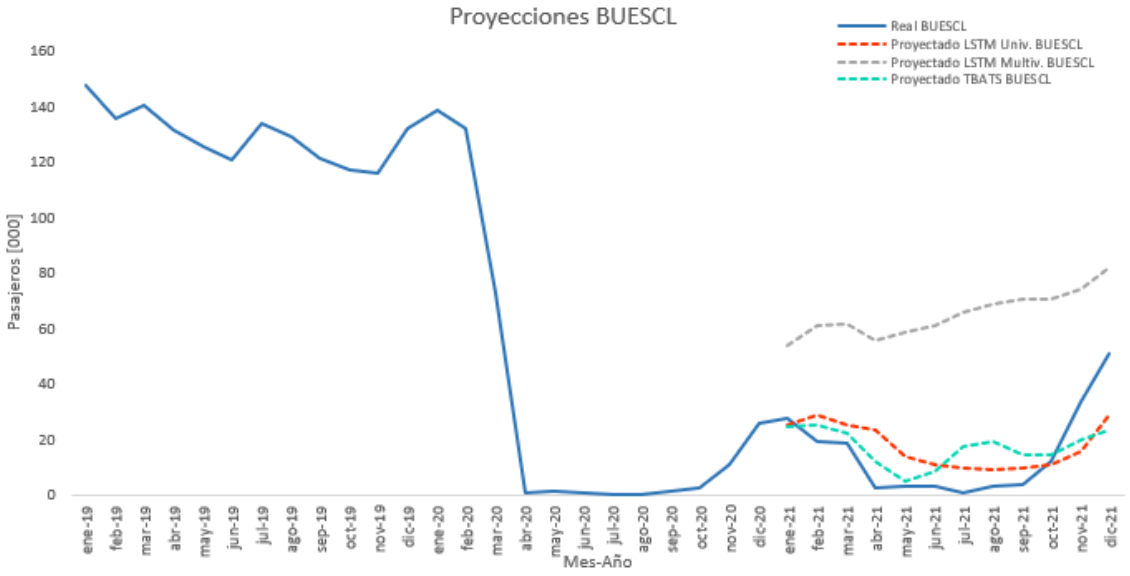
Se entrenaron y utilizaron los mismos modelos para proyectar 2021 y comparar nuevamente el desempeño de los modelos, tomando como medida de comparación entre modelos al RSME. Se observa que, en comparación a los resultados obtenidos para las proyecciones del año 2020, todos los modelos mejoran el RMSE. Por otra parte, la red LSTM univariada se mantiene como la de mejor desempeño en comparación a la red LSTM multivariada y el modelo TBATS, en casi todas las rutas del presente análisis.

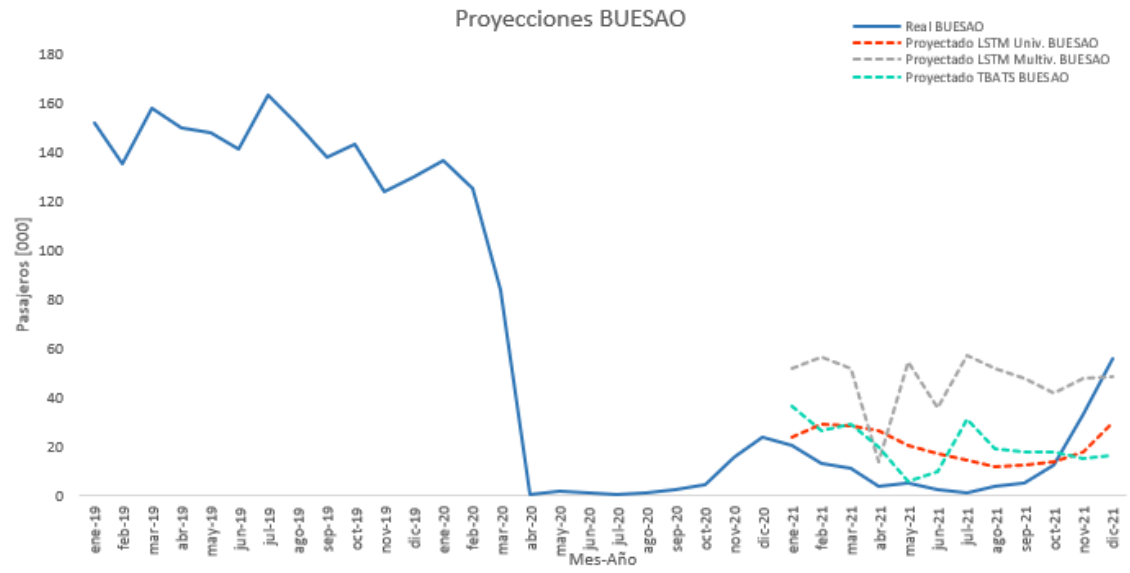
Tabla 6. Proyecciones 2021: RMSE abierto por ruta

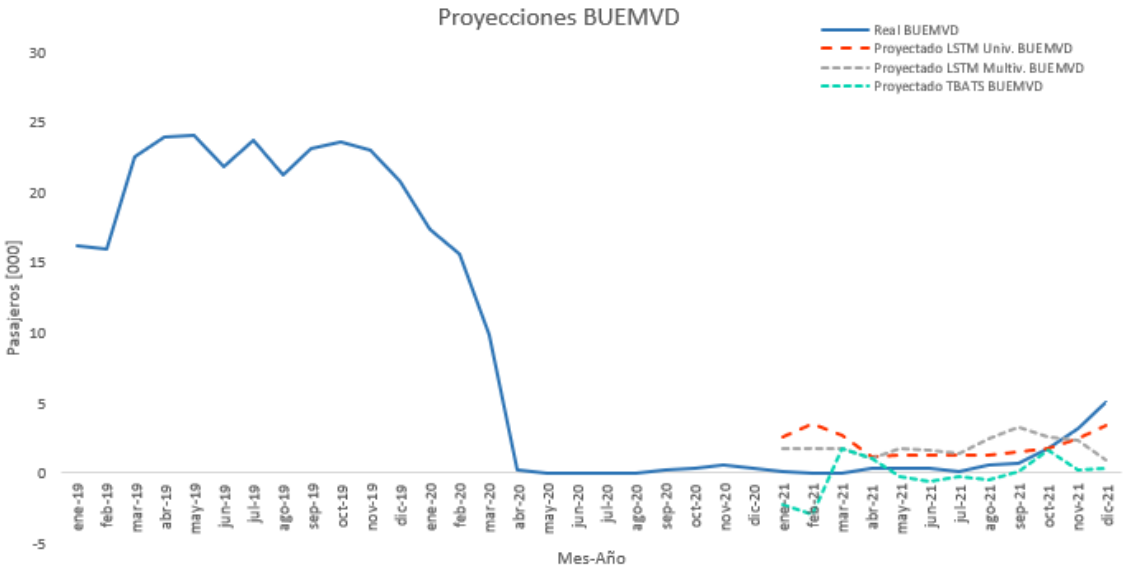
Ruta	RMSE Proyección 2021		
	LSTM Univariada	LSTM Multivariada	TBATS
BUESCL	12.0	65.6	12.3
BUESAO	14.9	47.9	18.8
BUEMVD	1.7	1.9	2.1
BUESRZ	2.8	14.2	3.8
BUELIM	4.1	34.5	3.6
BUEASU	2.3	4.9	3.0
BUERIO	3.4	33.8	9.2
BUEPDP	4.1	5.9	9.4
BUEFLN	1.6	13.9	7.0
BUESSA	0.6	3.5	2.4
BUEPOA	0.1	0.4	1.3

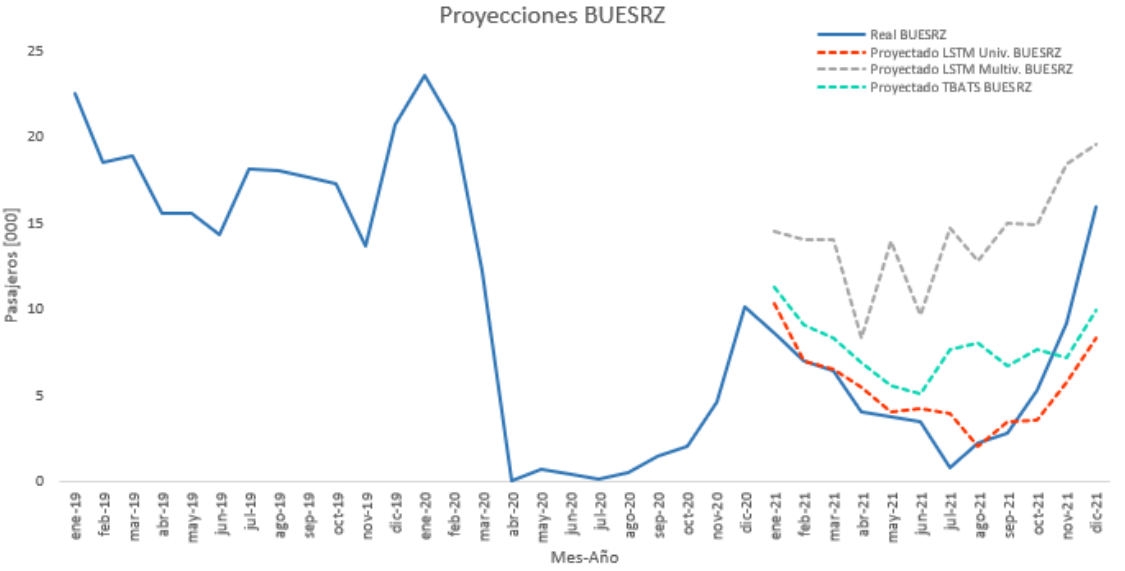
Cabe destacar, que el modelo TBATS, al incorporar el año 2020 dentro de los datos de entrenamiento, genera proyecciones negativas en algunas rutas, como BUEMVD, BUERIO, BUEPDP y BUEPOA. Las redes LSTM, nuevamente, demuestran su potencial para proyectar series de tiempo y su capacidad de incorporar la historia reciente como input de la proyección. A continuación, se exponen las proyecciones de los tres modelos implementados en el presente trabajo para los pasajeros de 2021.

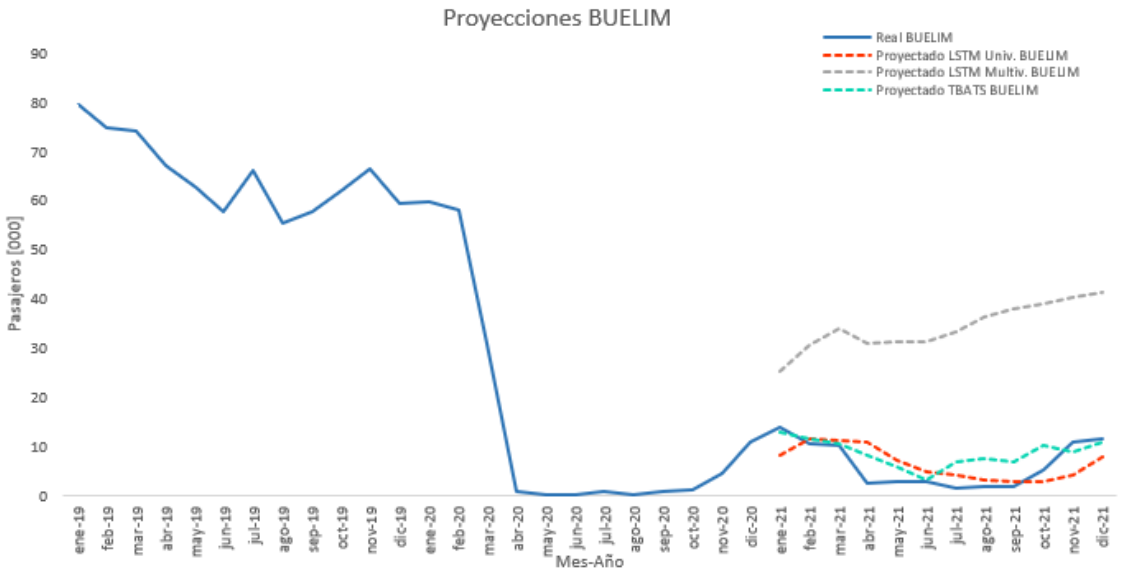
Figura 15. Proyecciones para 2021: Red LSTM Univariada, Multivariada y Modelo TBATS

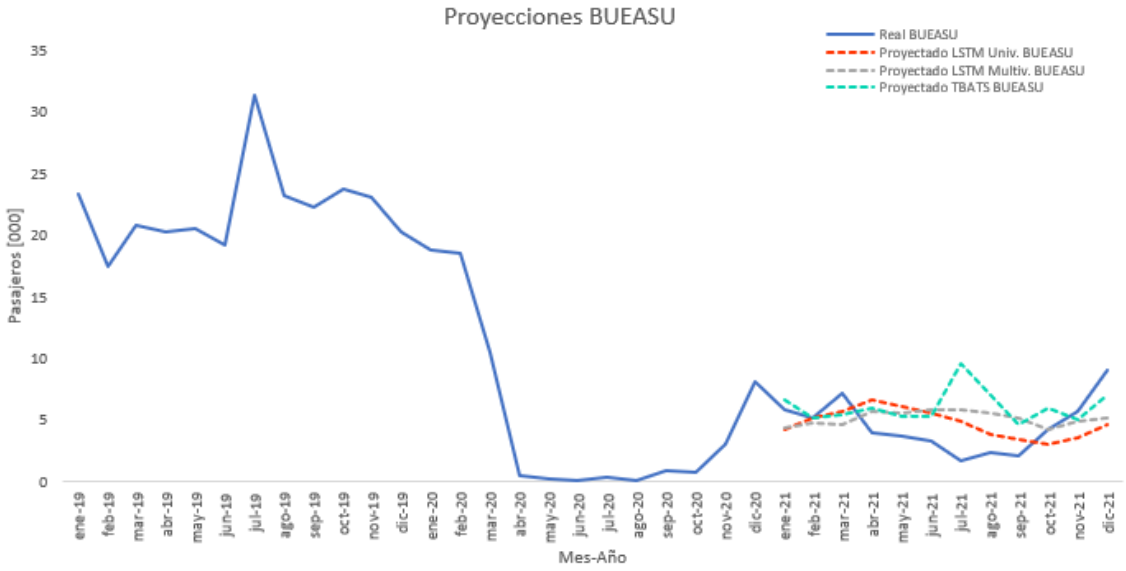

Proyecciones BUESCL
Real BUESCL
Proyectado LSTM Univ. BUESCL
Proyectado LSTM Multiv. BUESCL
Proyectado TBATS BUESCL
Pasajeros [000]
Mes-Año

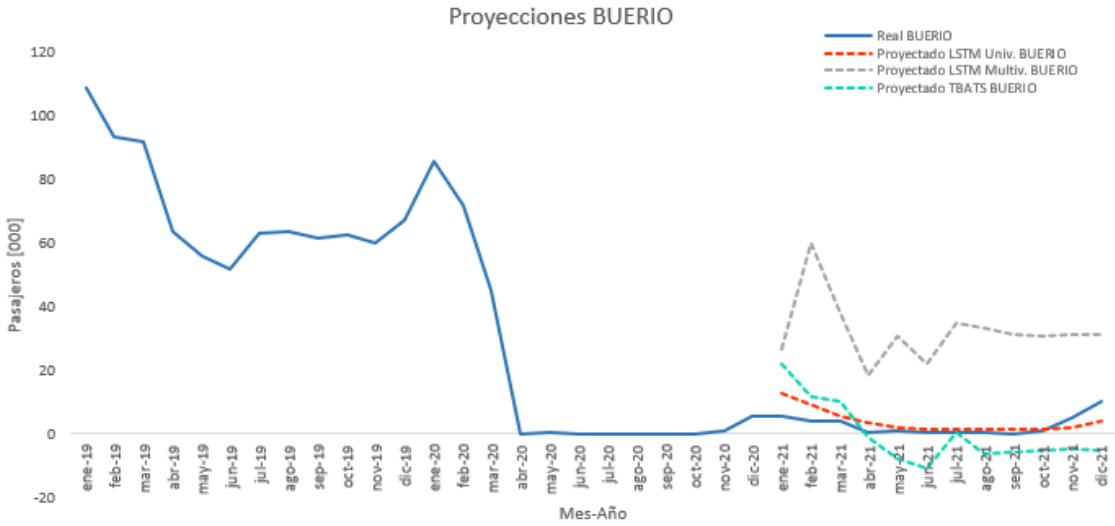

Proyecciones BUESAO
Real BUESAO
Proyectado LSTM Univ. BUESAO
Proyectado LSTM Multiv. BUESAO
Proyectado TBATS BUESAO
Pasajeros [000]
Mes-Año

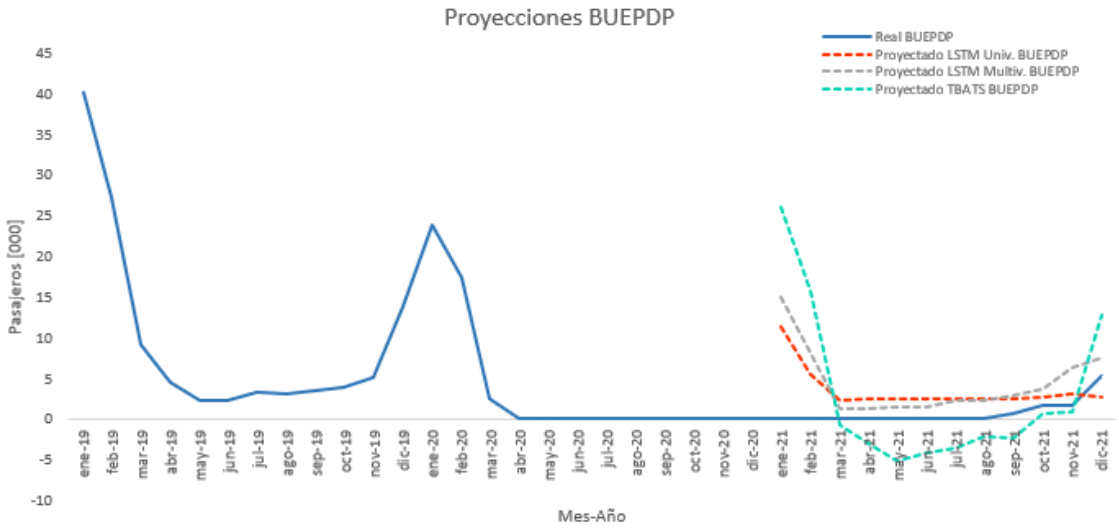

Proyecciones BUEMVD
Real BUEMVD
Proyectado LSTM Univ. BUEMVD
Proyectado LSTM Multiv. BUEMVD
Proyectado TBATS BUEMVD
Pasajeros [000]
Mes-Año

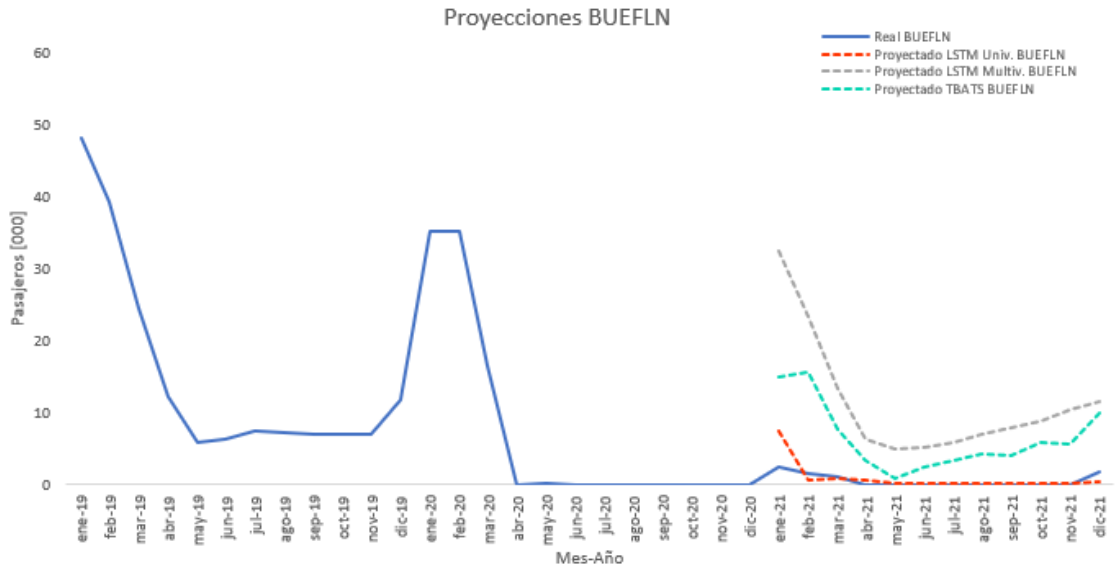

Proyecciones BUESRZ
Real BUESRZ
Proyectado LSTM Univ. BUESRZ
Proyectado LSTM Multiv. BUESRZ
Proyectado TBATS BUESRZ
Pasajeros [000]
Mes-Año

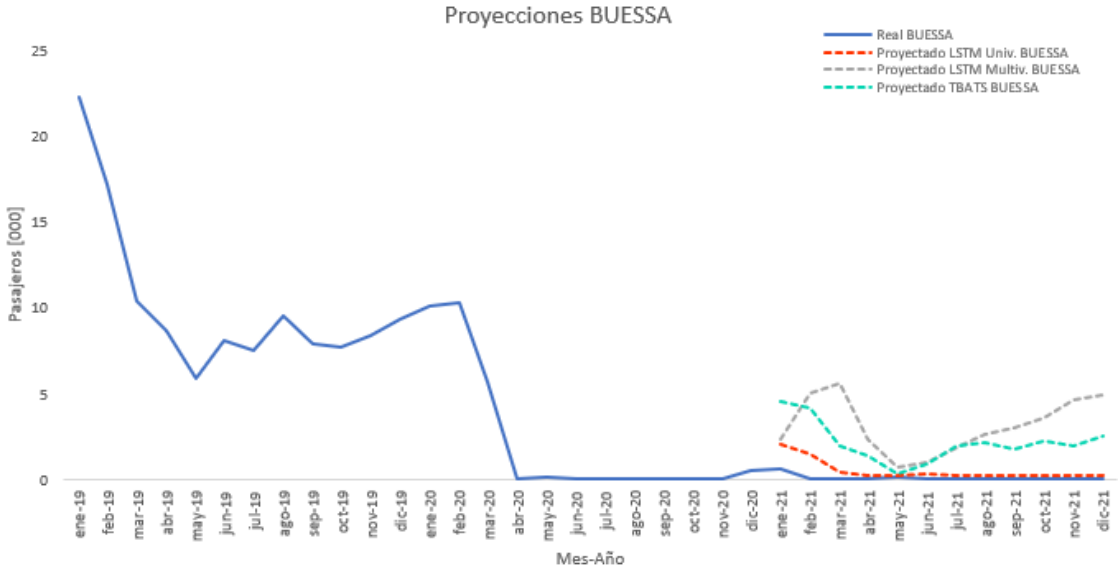

Proyecciones BUELIM
Real BUELIM
Proyectado LSTM Univ. BUELIM
Proyectado LSTM Multiv. BUELIM
Proyectado TBATS BUELIM
Pasajeros [000]
Mes-Año

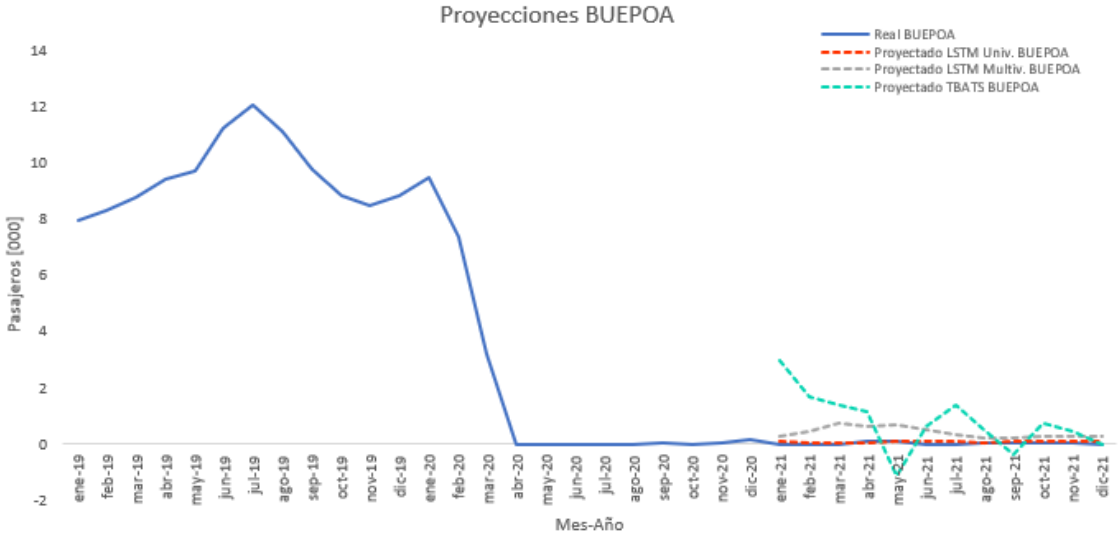

Proyecciones BUEASU
Real BUEASU
Proyectado LSTM Univ. BUEASU
Proyectado LSTM Multiv. BUEASU
Proyectado TBATS BUEASU
Pasajeros [000]
Mes-Año


Proyecciones BUERIO
Real BUERIO
Proyectado LSTM Univ. BUERIO
Proyectado LSTM Multiv. BUERIO
Proyectado TBATS BUERIO
Pasajeros [000]
Mes-Año


Proyecciones BUEPDP
Real BUEPDP
Proyectado LSTM Univ. BUEPDP
Proyectado LSTM Multiv. BUEPDP
Proyectado TBATS BUEPDP
Pasajeros [000]
Mes-Año







Fuente: Elaboración propia en base a datos de ANAC y proyecciones propias

4.3. Predicciones para el año 2022 y aplicación de negocio

A raíz de los resultados obtenidos en el entrenamiento y testeo de los modelos explorados en el presente trabajo, se comprobó que la red LSTM Univariada es la que mejor desempeño mostró tanto en los ejercicios de proyección del año 2020 como el del año 2021. Por esta razón se procede a realizar las predicciones para el año 2022 con este modelo.

Cabe aclarar que en los ejercicios de entrenamiento de los modelos se utilizaron los años 2020 y 2021 como muestras de validación para los modelos correspondientes (es decir, para el ejercicio de proyección del año 2020 se utilizaron los datos de dicho año como conjunto de validación, y el mismo ejercicio se aplicó para el año 2021). Luego, se utiliza el año 2022 como muestra *"out of time"* para las predicciones con el modelo seleccionado con mejor desempeño.

En primer lugar, se entrenó la red nuevamente con todos los datos disponibles, desde 2001 a 2021. Cabe destacar, que en cada ejercicio de predicción se utiliza como *input* los datos de los últimos 12 meses y se predice el mes siguiente. Luego, se incorpora dicha predicción dentro de la secuencia de *input* y se predice el mes siguiente, y así hasta completar doce meses de predicción. Por ejemplo, para predecir Enero 2022 se tiene como *input* los datos reales del año 2021:

[Ene21, Feb21, Mar21, Abr21, May21, Jun21, Jul21, Ago21, Sep21, Oct21, Nov21, Dic21]

Una vez realizado este primer paso de predicción, se incorpora la predicción de Enero 2022 como *input* para realizar la predicción de Febrero 2022, siendo el *input*:

[Feb21, Mar21, Abr21, May21, Jun21, Jul21, Ago21, Sep21, Oct21, Nov21, Dic21, Ene22]

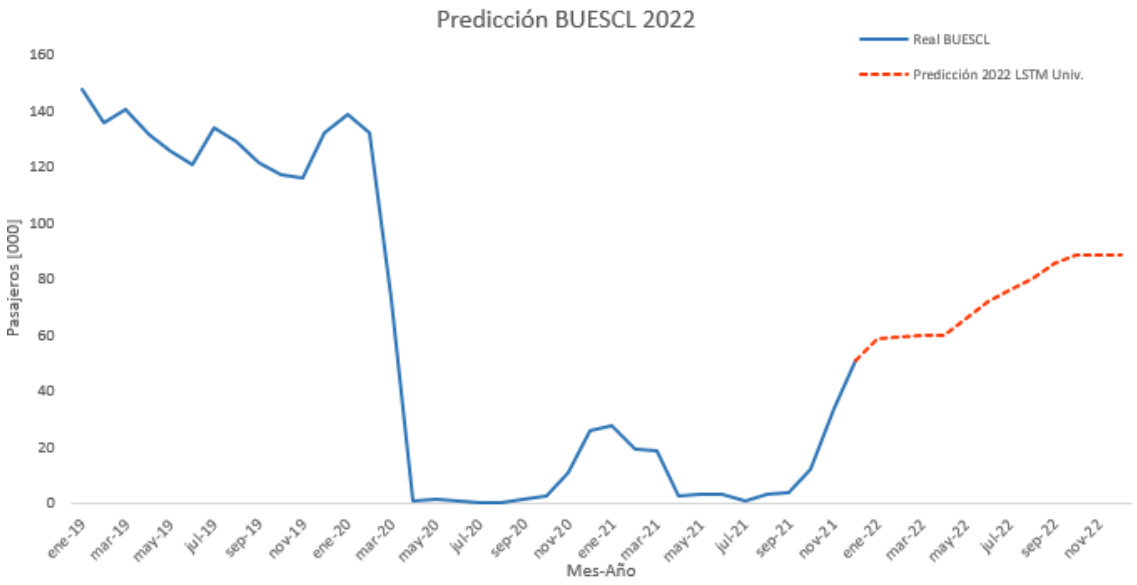
Para realizar esto, se definió la función *predict* que realiza el ejercicio de ir incorporando las predicciones como parte del *input* para la siguiente predicción. Se definen dos variables adicionales, que son *look_back* (*time_steps* en los ejercicios de proyección) que toma valor 12 y *num_prediction* que define la cantidad de periodos a predecir, en este caso, 12 meses, por lo que también toma valor 12. Por último, *model* contiene al modelo entrenado con todos los datos disponibles.

```
look_back = 12
def predict(num_prediction, model_BUESCL):
    prediction_list = df[-look_back:]

    for _ in range(num_prediction):
        x = prediction_list[-look_back:]
        x = x.reshape((1, look_back, 1))
        out = model.predict(x)[0][0]
        prediction_list = np.append(prediction_list, out)
    prediction_list = prediction_list[look_back-1:]

    return prediction_list

num_prediction = 12
forecast = predict(num_prediction, model)
```

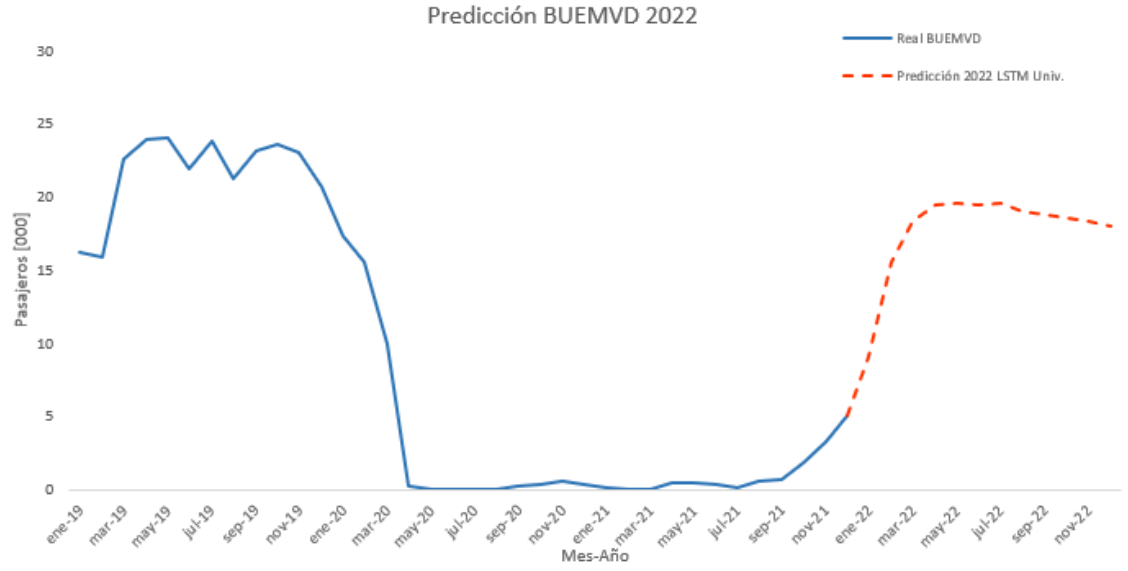
A partir de esto, se realizaron las predicciones para el año 2022, abiertas por mes, para las rutas contempladas en el presente trabajo. De estas predicciones se observa una recuperación de la cantidad de pasajeros regionales para el año 2022 en la mayoría de las rutas analizadas en el presente estudio.

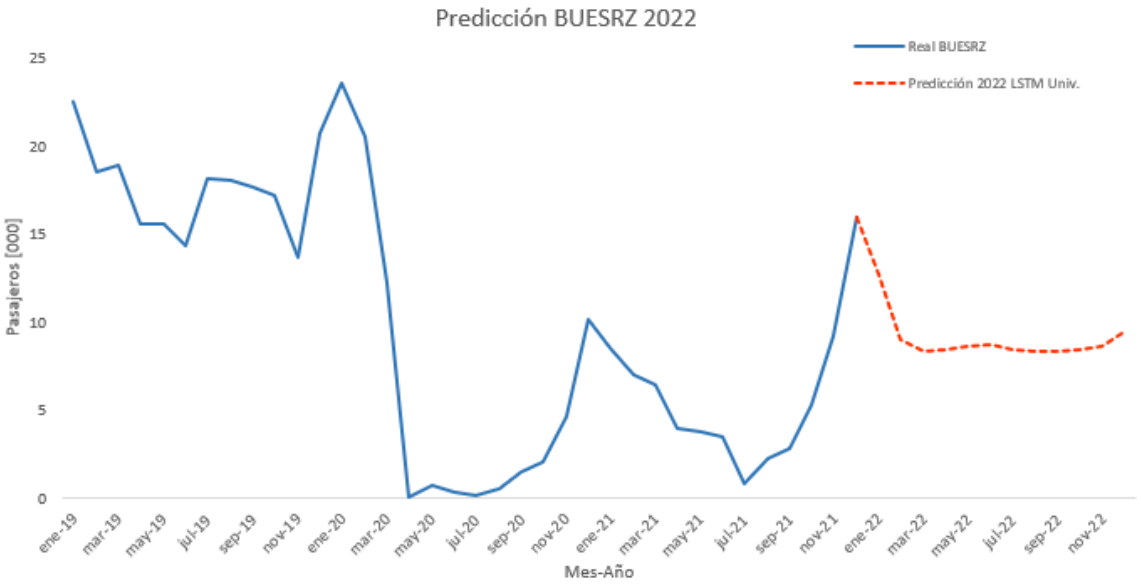
A continuación, se detallan los resultados a nivel ruta.

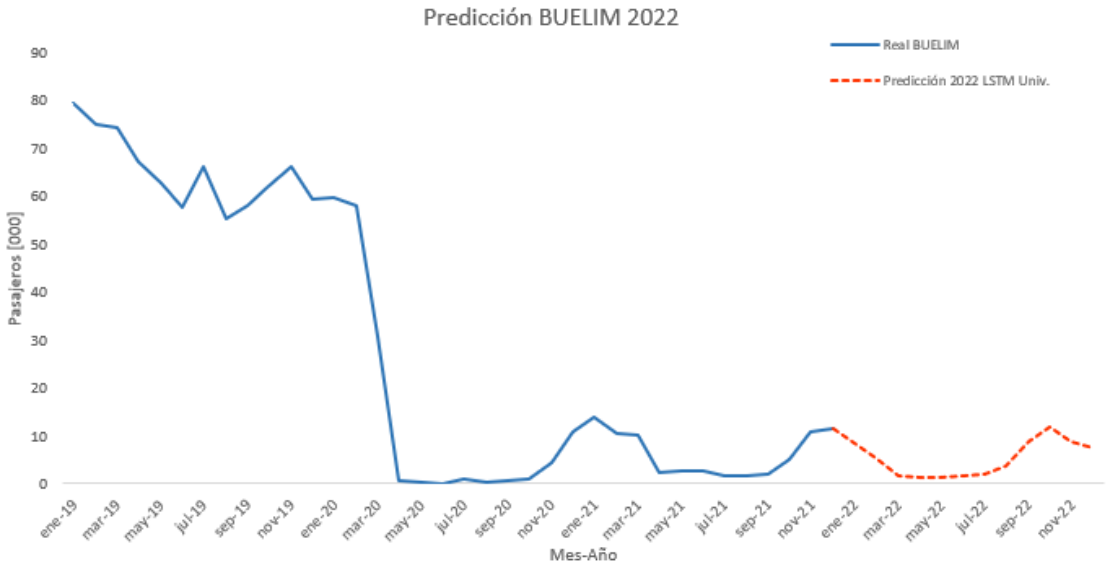
Figura 16. Predicciones año 2022 con la Red LSTM Univariada abierto por ruta

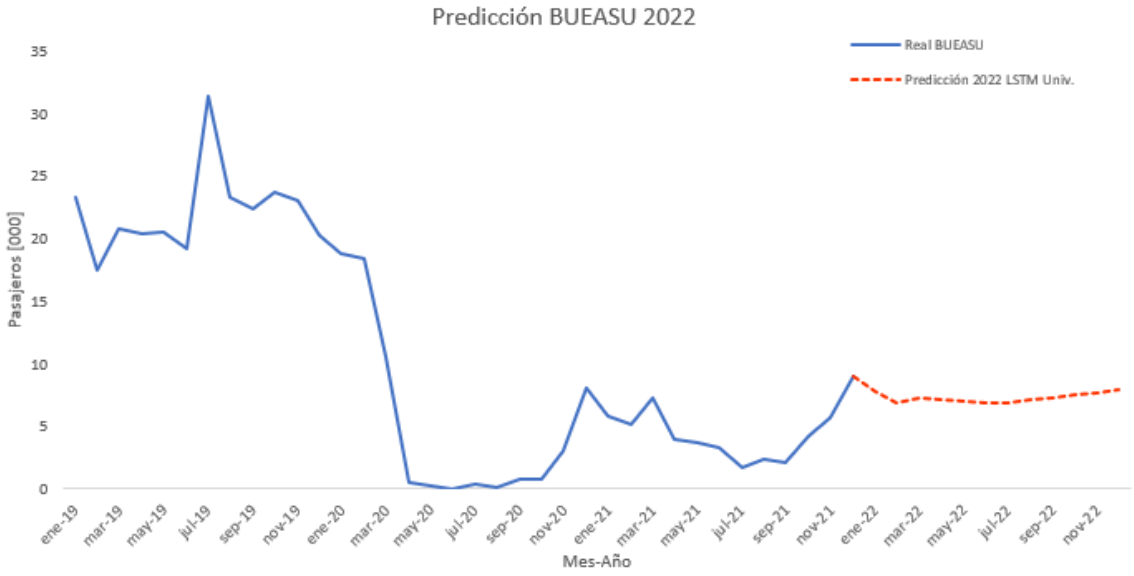



Predicción BUESAO 2022
Real BUESAO
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año

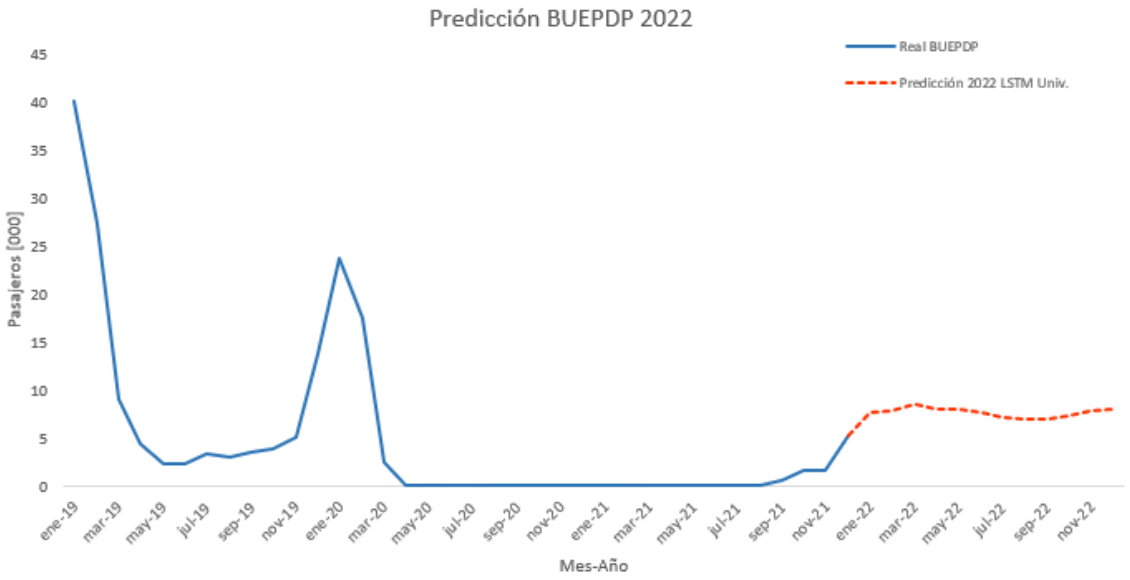

Predicción BUEMVD 2022
Real BUEMVD
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año

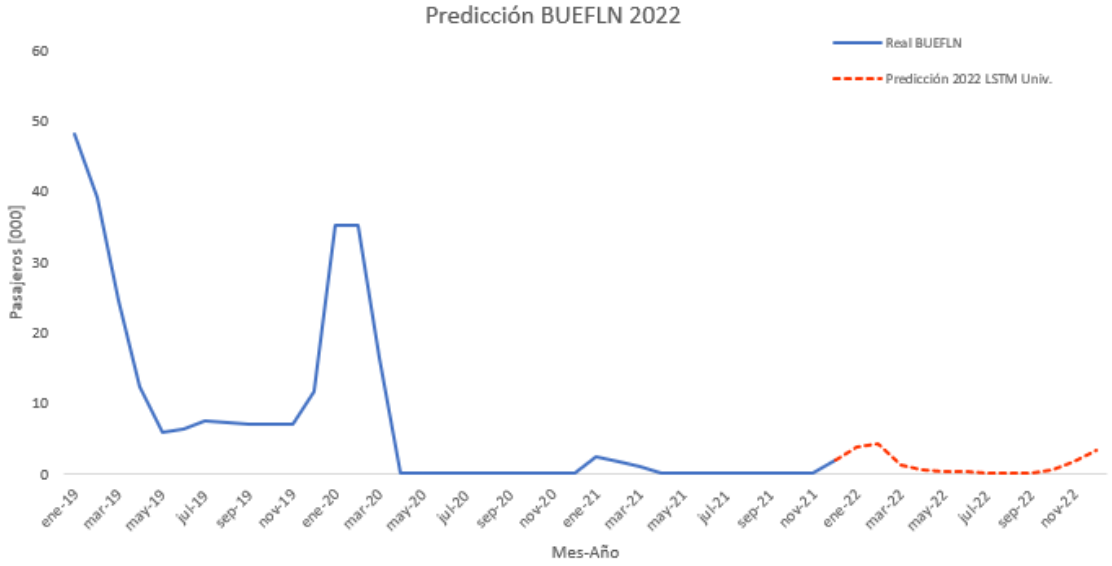

Predicción BUESRZ 2022
Real BUESRZ
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año

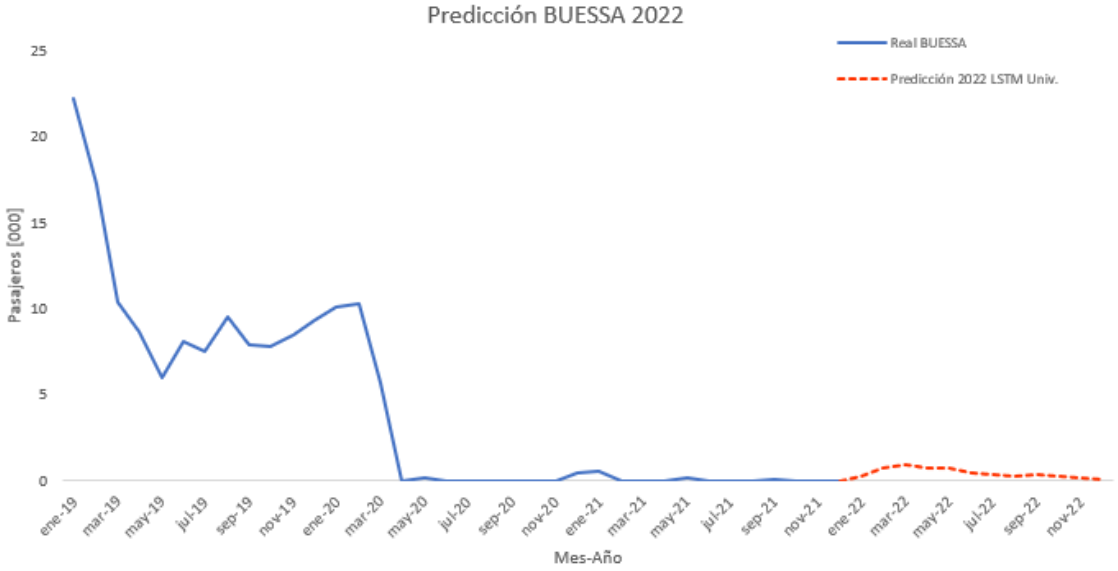

Predicción BUELIM 2022
Real BUELIM
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año

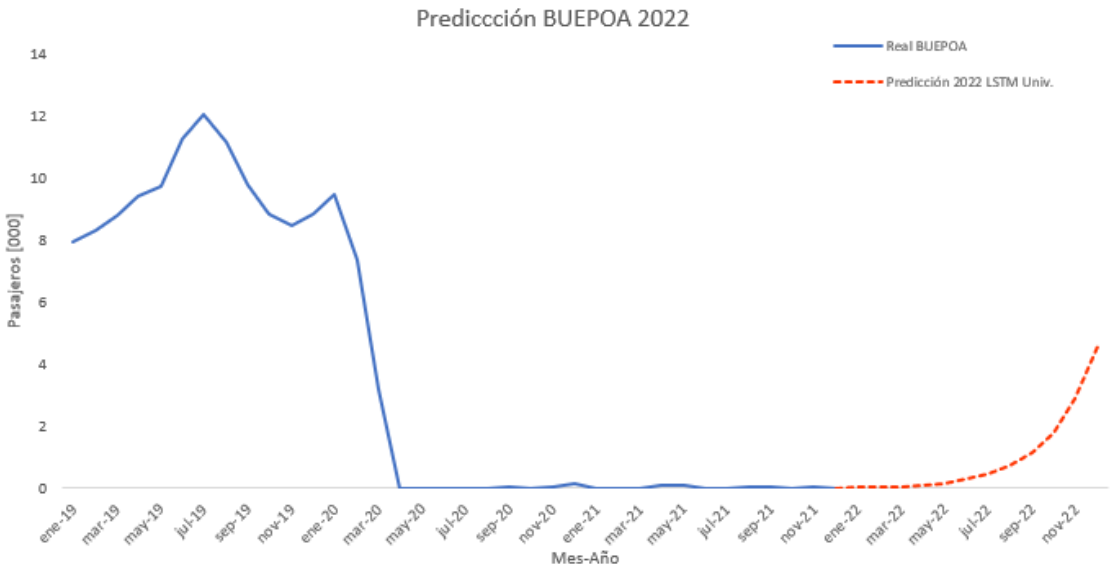

Predicción BUEASU 2022
Real BUEASU
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año


Predicción BUERIO 2022
Real BUERIO
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año


Predicción BUEPDP 2022
Real BUEPDP
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año


Predicción BUEFLN 2022
Real BUEFLN
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año


Predicción BUESSA 2022
Real BUESSA
Predicción 2022 LSTM Univ.
Pasajeros [000]
Mes-Año



Fuente: Elaboración propia en base a datos de ANAC y predicciones propias

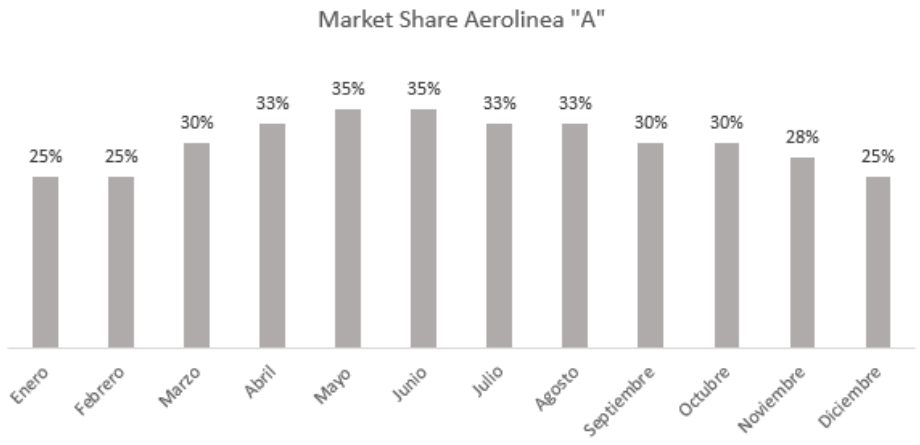
Aplicación de Negocio

Como se menciona a lo largo del presente trabajo, el propósito es poder brindar información pública a las aerolíneas, que operen o no las rutas analizadas, generando cierta previsibilidad a nivel mercado sobre que la cantidad de pasajeros esperados por mes. A partir de estas predicciones cualquier aerolínea puede incorporar esta información como input en la toma de decisiones para definir la estrategia de *revenue management* y oferta de vuelos.

Para ello, se debe aplicar un factor de *market share*, real o *target*, de la aerolínea que desee utilizar esta información. Por ejemplo, tomaremos como supuesto que la Aerolínea "A" que opera en el mercado regional en Argentina, desea incorporar la información brindada por el presente trabajo para algunas de sus rutas, por ejemplo, para la ruta BUELIM. Esta ruta únicamente opera desde el aeropuerto de Ezeiza, por lo que a priori, no se debe aplicar la separación de aeropuertos en la información.

El mercado de BUELIM es un mercado competido, con varias aerolíneas operando. Se define al *market share* como la porción de mercado o cantidad de pasajeros transportados por una aerolínea en un mes. La Aerolínea "A" presenta la siguiente evolución del *market share* abierta por mes.

Figura 17. Market share BUELIM 2021 Aerolínea “A”



Fuente: Elaboración propia

La aerolínea “A” desea mantener ese nivel de *market share* para 2022, y en base a las predicciones para ese año realizadas por el presente trabajo, definir su *estrategia revenue management* y oferta de vuelos alineada a ese objetivo de mercado.

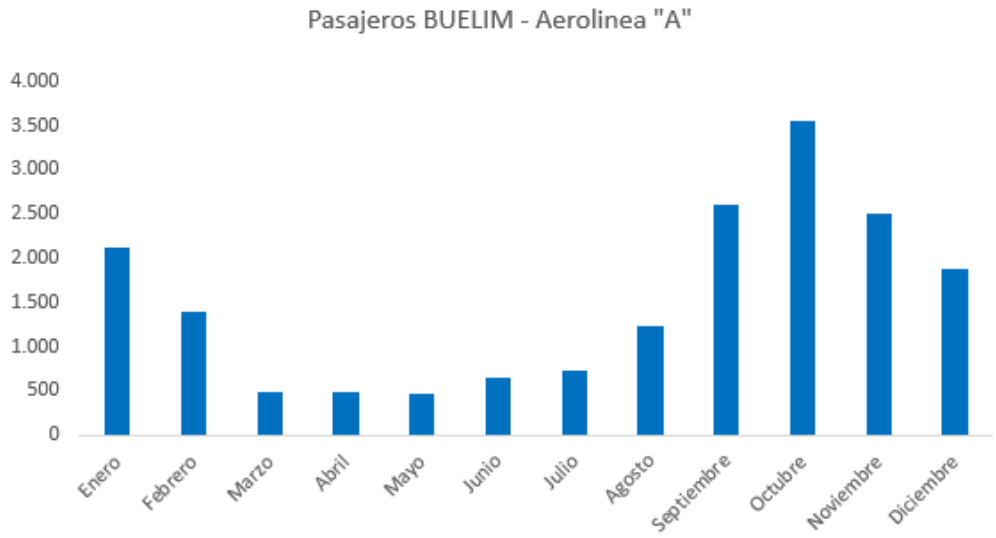
Las predicciones de pasajeros para BUELIM a nivel mercado son las siguientes.

Figura 18. Predicción de pasajeros BUELIM 2022

Fuente: Elaboración propia

Aplicando el *market share* de la aerolínea en cuestión a los pasajeros esperados para el 2022 se puede obtener el *target* de pasajeros que dicha aerolínea debería buscar para mantener su posición y objetivo de mercado en BUELIM en 2022.

Figura 19. Pasajeros BUELIM Aerolínea “A”



Fuente: Elaboración propia

En base a estos valores, la Aerolínea A puede incorporarlos dentro de sus proyecciones de demanda, haciendo un *forecast pooling* con sus estimaciones privadas. Luego con los pasajeros esperados, determinar su estrategia de oferta de vuelos y *revenue management* en pos de obtener ese volumen y maximizar sus beneficios.

5. Conclusiones

Como se menciona al inicio de este trabajo, para el sector aerocomercial resulta vital contar con información sobre la demanda futura de vuelos, de modo de gestionar eficientemente sus recursos y elaborar estrategias de *revenue management* que optimicen la flota y los asientos a vender.

Si bien existe información brindada por organismos públicos, como la ANAC, sobre los pasajeros volados por mes para todas las rutas operadas en el país, actualmente no se cuenta con predicciones agregadas sobre cantidad de pasajeros futuros por parte de estos organismos. Por ello, la necesidad y la pregunta que este trabajo intenta resolver es: ¿Qué nivel de pasajeros se espera en las principales rutas regionales de Argentina para los próximos doce meses? ¿Qué utilidad les pueden dar las aerolíneas y el sector aeronáutico a dichas proyecciones?

La respuesta a estas preguntas puede ser de gran utilidad para las aerolíneas que se encuentran operando dichas rutas o bien para cualquier aerolínea que evalúe entrar al mercado. Pero no resulta una tarea fácil, sobre todo considerando el contexto reciente, con la Pandemia COVID-19, que afectó drásticamente al sector aerocomercial y económico alrededor del globo.

A raíz de esto, se propuso explorar e implementar en el presente trabajo diferentes algoritmos o modelos para predecir pasajeros regionales a doce meses (un año) a nivel origen y destino, y entender cuál de ellos presenta el mejor desempeño (menor error en la estimación, con la métrica de error elegida) a la hora de incorporar estos cambios abruptos sucedidos en los últimos años. Para la implementación de los modelos se utilizó un código escrito en Python y se definieron, a criterio, algunos hiperparámetros para los modelos en el ejercicio de proyección (cantidad de neuronas, capas ocultas, épocas y tamaño de lote de entrenamiento, entre otros).

Como resultado se encontró que todos los modelos presentaban errores elevados en promedio en los ejercicios de proyección, pero la red LSTM univariada demostró un mejor desempeño, incorporando el contexto de caída abrupta en el año 2020 y leve recuperación durante 2021, por lo que se seleccionó dicho modelo para realizar las predicciones para el año 2022.

Las predicciones para el año 2022 muestran una recuperación del sector aeronáutico y del nivel de pasajeros regionales en Argentina, en un mercado completamente reversionado luego de la Pandemia. Con estas predicciones, se explicó la utilidad o aplicación de negocio de los datos obtenidos, aplicando un *market share* teórico sobre los pasajeros para brindar un *input* a la aerolínea ejemplificada, que puede ser utilizado en sus decisiones estratégicas de oferta de vuelos y *revenue management*.

A raíz del presente trabajo surgen distintas líneas futuras de investigación, posibles recomendaciones y mejoras para poder robustecer los modelos y generar estimaciones con mayor precisión y calidad para ser consumidas por las aerolíneas. Entre estas se encuentran la posibilidad de automatizar y evaluar múltiples hiperparámetros en el entrenamiento de los modelos, que permitan encontrar la combinación óptima que reduzca el error de estimación del modelo. Otra mejora podría relacionarse a la automatización de la extracción de los datos, aplicando técnicas de web scraping para descargar automáticamente las bases de ANAC una vez publicadas, a principios de cada mes, y realizar todas las transformaciones en Python.

Asimismo, incorporar variables alternativas en la red LSTM multivariada, dado que las utilizadas en el presente estudio no resultaron ser muy significativas en la proyección. Por último, una vez

mejorada la precisión del modelo, realizar el ejercicio de predicción a mayor plazo, un horizonte mas de mediano o largo plazo, como pueden ser dos o tres año. Esto ayudaría mucho a las aerolíneas a tener previsibilidad y mejorar aún más sus decisiones estratégicas. A su vez, cada aerolínea podría implementar la misma metodología valiéndose de sus propios datos e incorporando la tarifa como uno de los principales inputs económicos que no pudieron ser considerados en este trabajo.

Referencias

[1] Administración Nacional de Aviación Civil (2021). *Tabla 20: Pasajeros Comerciales Internacionales.* Estadísticas del mercado aerocomercial*. Tablas de Movimientos y Pasajeros 2001-2018 y Tablas de Movimientos y Pasajeros 2019-2021.* Recuperado de: https://datos.anac.gob.ar/estadisticas/

[2] Administración Nacional de Aviación Civil (2021). *Tabla 22: Vuelos Comerciales Internacionales.* Estadísticas del mercado aerocomercial*. Tablas de Movimientos y Pasajeros 2001-2018 y Tablas de Movimientos y Pasajeros 2019-2021.* Recuperado de: https://datos.anac.gob.ar/estadisticas/

[3] Administración Nacional de Aviación Civil (2018). Desregionalización de Aeroparque: se traspasa el 50% de los vuelos a Ezeiza. Recuperado de: https://www.anac.gov.ar/anac/web/index.php/1/1779/noticias-y-novedades/desregionalizacin-de-aeroparque-se-traspasa-el-50-de-los-vuelos-a-ezeiza

[4] Administración Nacional de Aviación Civil (2020). Aeroparque volverá a operar vuelos regionales. Recuperado de: https://www.anac.gov.ar/anac/web/index.php/1/2049/noticias-y-novedades/-ltstrong-gtaeroparque-volver-a-operar-vuelos-regionales-ltstrong-gt

[5] Frank La Vigne (2019*). ¿Cómo aprenden las redes neuronales?*. Microsoft Documentation. Recuperado de: https://docs.microsoft.com/es-es/archive/msdn-magazine/2019/april/artificially-intelligent-how-do-neural-networks-learn#:~:text=Cada%20ciclo%20de%20correcci%C3%B3n%20de,o%20%22minimizar%20la%20p%C3%A9rdida%22

[6] George V Jose (2019). Predicting Sequential Data using LSTM: An Introduction. Towards Data Science. Recuperado de: https://towardsdatascience.com/time-series-forecasting-with-recurrent-neural-networks-74674e289816

[7] Grzegorz Skorupa (2019). Forecasting Time Series with Multiple Seasonalities using TBATS in Python. Medium. Recuperado de: https://medium.com/intive-developers/forecasting-time-series-with-multiple-seasonalities-using-tbats-in-python-398a00ac0e8a

[8] Gupta,V., Sharma, K. and Sangwan, M.S. (2019) *Airlines passenger forecasting using LSTM based recurrent neural networks.* International Journal "Information Theories and Applications", Vol. 26, Number 2, pp 178-187.

[9] Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice.* 3rd edition, OTexts: Melbourne, Australia.

[10] Instituto Nacional de Estadísticas y Censos (2021). *Estimador mensual de actividad económica (EMAE).* Recuperado de: https://www.indec.gob.ar/indec/web/Nivel4-Tema-3-9-48

[11] International Air Transport Association (IATA) (2022). *IATA Airline and Location Codes*. Recuperado de: https://www.iata.org/en/services/codes/

[12]Jason Brownlee (2016). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Deep Learning for Time Series Forecasting. Recuperado de: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[13]Jason Brownlee (2017). Multivariate Time Series Forecasting with LSTMs in Keras. Deep Learning for Time Series Forecasting. Recuperado de: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

[14]Lawrence R. Weatherford (2003), *Neural Network forecasting for airlines: A comparative analysis.* Journal of Revenue and Pricing Management, Volume 1, Issue 4, pp 319–331.

[15]M. M. Mohie El-Din, M. S. Farag and A. A. Abouzeid (2017). *Airline Passenger Forecasting in EGYPT (Domestic and International).* International Journal of Computer Application (0975-8887), Vol. 165, Issue.6.

[16]Na8 (2018). *Breve historia de las Redes Neuronales Artificiales.* Aprende Machine Learning. Recuperado de: https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/

[17]Nadeem (2021). *Time Series Forecasting using TBATS Model.* Medium. Recuperado de: https://medium.com/analytics-vidhya/time-series-forecasting-using-tbats-model-ce8c429442a9

[18]Quang Hung Do, Shih-Kuei Lo, Jeng-Fung Chen, Chi-Luan Le and Luong Hoang Anh (2020), Forecasting Air Passenger Demand: A Comparison of LSTM and SARIMA, Journal of Computer Science, Original Research Paper, 2020.

[19]Rossenblatt (1958). *The perceptron: a probabilistic model for information storage and organization in the brain*. Cornell Aeronautical Laboratory. Psychological Review. Vol. 65, No. 6, 1958.

[20]Sitio Oficial del Estado Argentino (2020). Vuelven los vuelos regionales a Aeroparque. Recuperado de: https://www.argentina.gob.ar/noticias/vuelven-los-vuelos-regionales-aeroparque

[21]Srisaeng, P., Baxter, G. and Wild, G. (2015). *Using an artificial neural network approach to forecast Australia's domestic passenger air travel demand*. World Review of Intermodal Transportation Research, Vol. 5, No. 3, 2015, pp 281-313.

[22]Venelin Valkov (2019). Time Series Forecasting with LSTMs using TensorFlow 2 and Keras in Python. Curiously. Recuperado de: https://curiously.com/posts/time-series-forecasting-with-lstms-using-tensorflow-2-and-keras-in-python/

[23]Venelin Valkov (2019). Demand Prediction with LSTMs using TensorFlow 2 and Keras in Python. Curiously. Recuperado de: https://curiously.com/posts/demand-prediction-with-lstms-using-tensorflow-2-and-keras-in-python/

Apéndice A. Detalle predicción de pasajeros para el año 2022

A continuación, se detallan las predicciones de pasajeros para el año 2022, expresados en miles para las once rutas analizadas en el presente estudio.

Mes	BUESCL	BUESAO	BUEMVD	BUESRZ	BUELIM	BUEASU	BUERIO	BUEPDP	BUEFLN	BUESSA	BUEPOA
ene-22	58.5	59.2	9.0	12.7	8.5	7.8	11.2	7.6	3.9	0.3	0.0
feb-22	59.2	58.3	15.6	9.0	5.5	6.9	9.5	7.8	4.1	0.7	0.0
mar-22	59.7	57.7	18.3	8.3	1.6	7.2	7.7	8.5	1.3	0.9	0.1
abr-22	59.9	59.9	19.5	8.4	1.4	7.1	6.1	8.0	0.5	0.7	0.1
may-22	65.9	64.1	19.5	8.6	1.3	7.0	3.4	8.1	0.3	0.7	0.2
jun-22	71.6	67.6	19.5	8.7	1.8	6.8	0.9	7.6	0.2	0.5	0.3
jul-22	75.8	75.3	19.5	8.4	2.2	6.8	0.1	7.2	0.1	0.4	0.5
ago-22	80.0	81.6	19.0	8.3	3.7	7.1	0.1	7.0	0.1	0.3	0.7
sep-22	85.6	86.2	18.8	8.3	8.7	7.3	2.6	7.0	0.1	0.3	1.1
oct-22	88.2	90.1	18.6	8.4	11.8	7.5	7.3	7.2	0.6	0.3	1.8
nov-22	88.6	93.3	18.3	8.6	8.9	7.7	9.9	7.9	1.7	0.2	2.9
dic-22	88.6	95.7	17.9	9.5	7.4	7.9	9.6	8.0	3.2	0.1	4.6

Apéndice B. Detalle del código utilizado en Python

El siguiente código implica trabajar con una base de datos ya consolidada y depurada. Asimismo, se encuentra desarrollado para una ruta genérica, con lo cual puede utilizarse no solamente para rutas regionales sino para cualquier ruta que se desee analizar.

```
# Comenzamos con las configuraciones generales y las librerías a
utilizar

import numpy as np
import pandas as pd
import seaborn as sns
from pylab import rcParams
import matplotlib.pyplot as plt
from matplotlib import rc
from sklearn.model_selection import train_test_split
from pandas.plotting import register_matplotlib_converters
%matplotlib inline
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.preprocessing.sequence import TimeseriesGenerator
import os
os.environ['KMP_DUPLICATE_LIB_OK']='True'

# Seteamos del espacio de trabajo, en este caso, un espacio local

os.getcwd()
os.chdir('C:/Users/Desktop/)
```

```
### RED LSTM UNIVARIADA
# Este proceso se realiza para las 11 rutas bajo análisis.
# Se replicó para el año 2020 (como horizonte de proyección),
utilizando como entrenamiento 2001 a 2018 y prueba 2019 a 2020.

# 1) Preprocesamiento de los datos
# Cargamos el dataset a utilizar, ya se encuentra consolidado a
nivel ciudad. Se descargaron los datos de las diferentes fuentes de
información y se consolidaron en un archivo único llamado "Database"

df = pd.read_excel('Database.xlsx', sheet_name='Dataset_Univ')

# Para la red univariada filtramos los datos, quedándonos con las
observaciones desde 2001 a 2021

df=df.iloc[:252,:]

# Filtramos los datos para la ruta a analizar BUEXXX

df_BUEXXX= pd.DataFrame(df, columns=['BUEXXX'])

# Separamos los datos de entrenamiento y prueba. Usamos 2001-2019
como entrenamiento y 2020-2021 como prueba

train_size= 228
train_BUEXXX, test_BUEXXX = df_BUEXXX.iloc[0:train_size],
df_BUEXXX.iloc[train_size:len(df_BUEXXX)]

# Revisamos que los datos hayan quedado correctamente separados

print(len(train_BUEXXX), len(test_BUEXXX))
print(train_BUEXXX)
print(test_BUEXXX)

# Escalamos los datos entre 0 y 1 con MinMaxScaler

scaler = MinMaxScaler(feature_range=(0, 1))

train_BUEXXX = scaler.fit_transform(train_BUEXXX)
test_BUEXXX = scaler.fit_transform(test_BUEXXX)

# Revisamos que los datos hayan quedado correctamente convertidos

print(train_BUEXXX)
print(test_BUEXXX)

# Preparamos la información para la red neuronal con la función
create_dataset. A partir de esta función creamos secuencias de
datos, definidas por los time steps definidos

def create_dataset(X, y, time_steps=1):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        v = X.iloc[i:(i + time_steps)].values
        Xs.append(v)
        ys.append(y.iloc[i + time_steps])
    return np.array(Xs), np.array(ys)
```

```
# Hacemos el reshape aplicando la función, llevamos los datos a un
formato [muestras, pasos del tiempo, caracteristicas]

time_steps = 12

X_train_BUEXXX, y_train_BUEXXX = create_dataset(train_BUEXXX,
train_BUEXXX.BUEXXX,time_steps)

X_test_BUEXXX, y_test_BUEXXX = create_dataset(test_BUEXXX,
test_BUEXXX.BUEXXX, time_steps)

# Corroboramos que la información quedó correctamente para ser
utilizada en la red neuronal

print(X_train_BUEXXX.shape, y_train_BUEXXX.shape)
print(X_test_BUEXXX.shape, y_test_BUEXXX.shape)

# Observamos las secuencias creadas para entrenamiento y testeo

X_train_BUEXXX[0]
y_train_BUEXXX[0]
X_train_BUEXXX[1]
y_train_BUEXXX[1]

# 2) Modelado
# Definimos el modelo a utilizar: red neuronal LSTM secuencial con 1
capa oculta y 100 neuronas,función de perdida entropía cruzada y
función de optimización Adam con learning rate 0.01

model = keras.Sequential()
model.add(keras.layers.LSTM(100,input_shape=(X_train_BUEXXX.shape[1]
,X_train_BUEXXX.shape[2])))
model.add(keras.layers.Dense(1))
model.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(0.01))

model.summary()

# 3) Entrenamiento del modelo
# Para el entrenamiento se definen 100 épocas y tamaño de lote de 42

history_BUEXXX = model.fit(
    X_train_BUEXXX, y_train_BUEXXX,
    epochs=100,
    batch_size=42,
    verbose=1,
    shuffle=False
)

# Graficamos la perdida en entrenamiento y prueba

plt.plot(history_BUEXXX.history['loss'], label='train')
plt.plot(history_BUEXXX.history['val_loss'], label='test')
plt.legend()
plt.show()

# 3) Evaluación del modelo
# Hacemos las proyecciones con el modelo entrenado

y_pred_BUEXXX = model.predict(X_test_BUEXXX)
```

```
# Re-escalamos los datos para poder calcular el error del modelo

y_train_BUEXXX_inv =
scaler.inverse_transform(np.expand_dims(y_train_BUEXXX,axis=0)).flat
ten()
y_test_BUEXXX_inv =
scaler.inverse_transform(np.expand_dims(y_test_BUEXXX,axis=0)).flatt
en()
y_pred_BUEXXX_inv =
scaler.inverse_transform(y_pred_BUEXXX).flatten()

# Calculamos el RMSE en proyecciones

rmse_BUEXXX = np.sqrt(mean_squared_error(y_test_BUEXXX,
y_pred_BUEXXX_inv))

print('Test RMSE: %.3f' % rmse_BUEXXX)

# Graficamos las series originales y las proyecciones

plt.plot(np.arange(0, len(y_train_BUEXXX)), y_train_BUEXXX, 'grey',
label="history")
plt.plot(np.arange(len(y_train_BUEXXX), len(y_train_BUEXXX) +
len(y_test_BUEXXX)), y_test_BUEXXX,'darkblue', label="real")
plt.plot(np.arange(len(y_train_BUEXXX), len(y_train_BUEXXX) +
len(y_test_BUEXXX)), y_pred_BUEXXX_inv, 'orangered',
label="prediction")
plt.title('BUEXXX Forecast')
plt.ylabel('Value')
plt.xlabel('Time Step')
plt.legend()
plt.show();

# Ajustamos los formatos de los datos para poder consolidar
resultados en un archivo final

y_pred_BUEXXX_inv=y_pred_BUEXXX_inv.reshape((12,))
y_pred_BUEXXX_inv_ru=pd.Series(y_pred_BUEXXX_inv)


# Una vez realizado el ejercicio para las 11 rutas en análisis,
guardamos los datos obtenidos con la red neuronal univariada

from pandas import concat

df_final_LSTM_UNIV=pd.concat([y_pred_BUESCL_inv_ru,y_pred_BUESAO_inv
_ru,y_pred_BUEMVD_inv_ru,y_pred_BUESRZ_inv_ru,y_pred_BUELIM_inv_ru,y
_pred_BUEASU_inv_ru,y_pred_BUERIO_inv_ru,y_pred_BUEPDP_inv_ru,y_pred
_BUEFLN_inv_ru,y_pred_BUESSA_inv_ru,y_pred_BUEPOA_inv_ru],axis=1)

df_final_LSTM_UNIV.to_excel('LTMS_UNIV_Proyecciones_2021.xlsx')

# Revisamos las proyecciones

print(df_final_LSTM_UNIV)

# Guardamos los RMSE obtenidos por la red univariada para comparar
los modelos
```

```
df_final_RSME_LSTM_univ=pd.concat([rmse_BUESCL,rmse_BUESAO,rmse_BUEM
VD,rmse_BUESRZ,rmse_BUELIM,rmse_BUEASU,rmse_BUERIO,rmse_BUEPDP,rmse_
BUEFLN,rmse_BUESSA,rmse_BUEPOA],axis=1)

df_final_RSME_LSTM_univ.to_excel('RMSE_LTMS_UNIV_Proyecciones_2021.x
lsx')

print(df_final_RSME_LSTM_univ)



### RED LSTM MULTIVARIADA
# Este proceso se realiza para las 11 rutas bajo análisis.
# Se replica para el año 2020 (como horizonte de proyección),
utilizando como entrenamiento 2001 a 2018 y prueba 2019 a 2020.

# 1) Preprocesamiento de los datos
# Cargamos el dataset, previamente transformado, a nivel ruta, para
las 11 rutas en análisis. Se agregaron también los datos de
evolución del tipo de cambio y EMAE

df_BUEXXX_lstm_multiv=pd.read_excel('Database.xlsx',
heet_name='Dataset_Multiv_BUEXXX', indexcol=0)

# Para la red multivariada filtramos los datos, quedándonos con las
observaciones desde 2004 a 2021

df_BUEXXX_lstm_multiv=df_BUEXXX_lstm_multiv.iloc[:216,1:]

# Separamos los datasets de entrenamiento y prueba: 2001-2019
entrenamiento y 2020-2021 prueba

train_size= 192

train_BUEXXX_lstm_multiv,test_BUEXXX_lstm_multiv=df_BUEXXX_lstm_mult
iv.iloc[0:train_size],
df_BUEXXX_lstm_multiv.iloc[train_size:len(df_BUEXXX)]

# Revisamos que los datos hayan quedado correctamente separados

print(len(train_BUEXXX_lstm_multiv), len(test_BUEXXX_lstm_multiv))
print(train_BUEXXX_lstm_multiv)
print(test_BUEXXX_lstm_multiv)

# Escalamos los datos entre 0 y 1 con MinMaxScaler

f_columns = ['Vuelos_BUEXXX', 'TC', 'EMAE']
f_col_scaler = MinMaxScaler(feature_range=(0, 1))
pax_scaler = MinMaxScaler(feature_range=(0, 1))

train_BUEXXX_lstm_multiv.loc[:,f_columns]=f_col_scaler.fit_transform
(train_BUEXXX_lstm_multiv[f_columns])
train_BUEXXX_lstm_multiv['Pax_BUEXXX']=pax_scaler.fit_transform(trai
n_BUEXXX_lstm_multiv[['Pax_BUEXXX']])

test_BUEXXX_lstm_multiv.loc[:,f_columns]=f_col_scaler.fit_transform(
test_BUEXXX_lstm_multiv[f_columns])
test_BUEXXX_lstm_multiv['Pax_BUEXXX']=pax_scaler.fit_transform(test_
BUEXXX_lstm_multiv[['Pax_BUEXXX']])

# Revisamos que los datos hayan quedado correctamente convertidos
```

```
print(train_BUEXXX_lstm_multiv)
print(test_BUEXXX_lstm_multiv)
# Preparamos la información para la red neuronal con la función
create_dataset. A partir de esta función creamos secuencias de
datos, definidas por los time steps definidos

def create_dataset(X, y, time_steps=1):
    Xs, ys = [], []
    for i in range(len(X) - time_steps):
        v = X.iloc[i:(i + time_steps)].values
        Xs.append(v)
        ys.append(y.iloc[i + time_steps])
    return np.array(Xs), np.array(ys)

# Hacemos el reshape aplicando la función, llevamos los datos a un
formato [muestras, pasos del tiempo, características]

time_steps = 12

X_train_BUEXXX_lstm_multiv,y_train_BUEXXX_lstm_multiv=create_dataset
(train_BUEXXX_lstm_multiv,
train_BUEXXX_lstm_multiv.BUEXXX,time_steps)

X_test_BUEXXX_lstm_multiv,y_test_BUEXXX_lstm_multiv=create_dataset(t
est_BUEXXX_lstm_multiv, test_BUEXXX_lstm_multiv.BUEXXX, time_steps)

# Corroboramos que la información es correcta

print(X_train_BUEXXX_lstm_multiv.shape,
y_train_BUEXXX_lstm_multiv.shape)

print(X_test_BUEXXX_lstm_multiv.shape,
y_test_BUEXXX_lstm_multiv.shape)

# Observamos las secuencias creadas para entrenamiento y testeo

X_train_BUEXXX_lstm_multiv[0]
y_train_BUEXXX_lstm_multiv[0]
X_train_BUEXXX_lstm_multiv[1]
y_train_BUEXXX_lstm_multiv[1]


# 2) Modelado
# Definimos el modelo a utilizar: red neuronal LSTM secuencial con 1
capa oculta y 100 neuronas, función de perdida entropía cruzada y
función de optimización Adam con learning rate 0.01

model = keras.Sequential()
model.add(keras.layers.LSTM(units=100,
input_shape=(X_train_BUEXXX_lstm_multiv.shape[1],
X_train_BUEXXX_lstm_multiv.shape[2])))
model.add(keras.layers.Dense(units=1))
model.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(0.01))

model.summary()

# 3) Entrenamiento del modelo
# Para el entrenamiento se definen 100 épocas y tamaño de lote de 42
```

```
history_BUEXXX_lstm_multiv = model.fit(
    X_train_BUEXXX_lstm_multiv, y_train_BUEXXX_lstm_multiv,
    epochs=100,
    batch_size=42,
    verbose=1,
    shuffle=False
)

# Graficamos la perdida en entrenamiento y testeo
plt.plot(history_BUEXXX_lstm_multiv.history['loss'], label='train')
plt.plot(history_BUEXXX_lstm_multiv.history['val_loss'],
label='test')
plt.legend()
plt.show()

# 4) Evaluación del modelo
# Hacemos las proyecciones con el modelo entrenado

y_pred_BUEXXX_lstm_multiv = model.predict(X_test_BUEXXX_lstm_multiv)

# Re-escalamos los datos para poder calcular el error

y_train_inv_BUEXXX_lstm_multiv =
pax_scaler.inverse_transform(y_train_BUEXXX_lstm_multiv.reshape(1, -
1))
y_test_inv_BUEXXX_lstm_multiv =
pax_scaler.inverse_transform(y_test_BUEXXX_lstm_multiv.reshape(1, -
1))
y_pred_inv_BUEXXX_lstm_multiv =
pax_scaler.inverse_transform(y_pred_BUEXXX_lstm_multiv)

# Revisamos que los datos hayan quedado correctamente

print(y_train_inv_BUEXXX_lstm_multiv)
print(y_test_inv_BUEXXX_lstm_multiv)
print(y_pred_inv_BUEXXX_lstm_multiv)

# Calculamos el RMSE en proyecciones

rmse_BUEXXX_lstm_multiv =
np.sqrt(mean_squared_error(y_test_BUEXXX_lstm_multiv,
y_pred_BUEXXX_inv_lstm_multiv))
print('Test RMSE: %.3f' % rmse_BUEXXX_lstm_multiv)

# Graficamos las series originales y las proyecciones para el año
2021

plt.plot(np.arange(0,len(y_train_BUEXXX_lstm_multiv)),y_train_inv_BU
EXXX_lstm_multiv.flatten(), 'g', label="history")
plt.plot(np.arange(len(y_train_BUEXXX_lstm_multiv),len(y_train_BUEXX
X_lstm_multiv)+len(y_test_BUEXXX_lstm_multiv)),y_test_inv_BUEXXX_lst
m_multiv.flatten(), marker='.', label="true")
plt.plot(np.arange(len(y_train_BUEXXX_lstm_multiv),len(y_train_BUEXX
X_lstm_multiv)+
len(y_test_BUEXXX_lstm_multiv)),y_pred_inv_BUEXXX_lstm_multiv.flatte
n(),'r',label="prediction")
plt.ylabel('Passengers BUEXXX')
plt.xlabel('Time Step')
plt.legend()
plt.show();
```

```
# Hacemos zoom sobre las proyecciones para 2021

plt.plot(y_test_inv_BUEXXX_lstm_multiv.flatten(), marker='.',
label="true")
plt.plot(y_pred_inv_BUEXXX_lstm_multiv.flatten(), 'r',
label="prediction")
plt.ylabel('Passengers BUEXXX')
plt.xlabel('Time Step')
plt.legend()
plt.show();

# Ajustamos los formatos de los datos para poder consolidar
resultados en un archivo final

y_pred_BUEXXX_inv_lstm_multiv=y_pred_BUEXXX_inv.reshape((12,))
y_pred_BUEXXX_inv_rm=pd.Series(y_pred_BUEXXX_inv_lstm_multiv)

# Una vez realizado el ejercicio para las 11 rutas en análisis,
guardamos los datos obtenidos con la red neuronal multivariada

from pandas import concat

df_final_LSTM_MULTIV=pd.concat([y_pred_BUESCL_inv_rm,y_pred_BUESAO_i
nv_rm,y_pred_BUEMVD_inv_rm,y_pred_BUESRZ_inv_rm,y_pred_BUELIM_inv_rm
,y_pred_BUEASU_inv_rm,y_pred_BUERIO_inv_rm,y_pred_BUEPDP_inv_rm,y_pr
ed_BUEFLN_inv_rm,y_pred_BUESSA_inv_rm,y_pred_BUEPOA_inv_rm],axis=1)
df_final_LSTM_MULTIV.to_excel('LTMS_MULTIV_Proyecciones_2021.xlsx')

# Revisamos las proyecciones
print(df_final_LSTM_MULTIV)

# Guardamos los RMSE obtenidos en prueba para comparar los modelos

df_final_RSME_LSTM_MULTIV=pd.concat([rmse_BUESCL_lstm_multiv,rmse_BU
ESAO_lstm_multiv,rmse_BUEMVD_lstm_multiv,rmse_BUESRZ_lstm_multiv,rms
e_BUELIM_lstm_multiv,rmse_BUEASU_lstm_multiv,rmse_BUERIO_lstm_multiv
,rmse_BUEPDP_lstm_multiv,rmse_BUEFLN_lstm_multiv,rmse_BUESSA_lstm_mu
ltiv,rmse_BUEPOA_lstm_multiv],axis=1)

df_final_RSME_LSTM_MULTIV.to_excel('RMSE_LTMS_MULTIV_Proyecciones_20
21.xlsx')

print(df_final_RSME_LSTM_MULTIV)



### MODELO TBATS
# Este proceso se realiza para las 11 rutas bajo análisis.
# Se replica para el año 2020 (como horizonte de proyección),
utilizando como entrenamiento 2001 a 2019 y como prueba 2020.

# 1) Preprocesamiento de los datos
# Cargamos el dataset a utilizar, ya se encuentra consolidado a
nivel ciudad.  Se descargaron los datos a utilizar de las diferentes
fuentes de información y se consolidaron en un archivo único llamado
"Database"

df = pd.read_excel('Database.xlsx', sheet_name='Dataset_Univ')

# Filtramos los datos, quedándonos con las observaciones desde 2001
a 2021
```

```
df=df.iloc[:252,:]

# Filtramos los datos para la ruta a analizar BUEXXX

y_BUEXXX= df['BUEXXX']

# Separamos los datasets de entrenamiento y prueba: 2001-2020
entrenamiento y 2021 prueba

y_to_train_BUEXXX_2021 = y_BUEXXX.iloc[:(len(y_BUEXXX)-12)]
y_to_test_BUEXXX_2021 = y_BUESCL.iloc[(len(y_BUEXXX)-12):]

# Revisamos que los datos hayan quedado correctamente separados

print(y_to_train_BUEXXX_2021)
print(y_to_test_BUEXXX_2021)

# 2) Modelado
# Definimos el modelo a utilizar con periodicidad mensual y anual

estimator = TBATS(seasonal_periods=(1,12))

# 3) Entrenamiento del modelo
# Para el entrenamiento utilizamos la función fit aplicando el
dataset de entrenamiento y el estimador definido

fitted_model_BUEXXX_2021 = estimator.fit(y_to_train_BUEXXX_2021)

# Vemos un resumen del modelo entrenado y las proyecciones en el
entrenamiento

print('\n\nSUMMARY FUNCTION\n\n')
print(fitted_model_BUEXXX_2021.summary())
print('\n\nIN SAMPLE PREDICTIONS\n\n')
print('Original time series (5 first values)',
fitted_model_BUEXXX_2021.y[:5])
print('Predictions (5 first values)',
fitted_model_BUEXXX_2021.y_hat[:5])
print('Residuals (5 first values)',
fitted_model_BUEXXX_2021.resid[:5])

# 4) Evaluación del modelo
# Hacemos las proyecciones con el modelo entrenado a 12 meses en
avance

steps=12

y_forecasted_BUEXXX_2021 =
fitted_model_BUEXXX_2021.forecast(steps=steps)

# Calculamos el RMSE en proyecciones y observamos los valores
proyectados

print('\n\nFORECAST\n\n')
print('Values', y_forecasted_BUEXXX_2021)
print('RMSE',
np.sqrt(mean_squared_error(y_forecasted_BUEXXX_2021,y_to_test_BUEXXX
_2021)))
RMSE_BUEXXX_2021_TBATS=np.sqrt(mean_squared_error(y_forecasted_BUEXX
X_2021,y_to_test_BUEXXX_2021))
```

```
# Graficamos las series originales y las proyecciones para el año
2021

x = np.arange(240, 252, 1)
train_window = 12


plt.title('BUEXXX Forecast')
plt.ylabel('Total Passengers')
plt.xlabel('Month')
plt.grid(False)
plt.autoscale(axis='x', tight=True)
plt.plot(y_BUEXXX)
plt.plot(x,y_forecasted_BUEXXX_2021, 'turquoise')
plt.show()

plt.title('BUEXXX Forecast')
plt.ylabel('Total Passengers')
plt.xlabel('Month')
plt.grid(False)
plt.autoscale(axis='x', tight=True)

plt.plot(y_BUEXXX[-train_window*2:])
plt.plot(x,y_forecasted_BUEXXX_2021, 'turquoise')
plt.show()

# Guardamos los datos obtenidos con el modelo TBATS en excel

from pandas import concat

df_final_TBATS=pd.concat([y_forecasted_BUESCL_2021,y_forecasted_BUES
AO_2021,y_forecasted_BUEMVD_2021,y_forecasted_BUESRZ_2021,y_forecast
ed_BUELIM_2021,y_forecasted_BUEASU_2021,y_forecasted_BUERIO_2021,y_f
orecasted_BUEPDP_2021,y_forecasted_BUEFLN_2021,y_forecasted_BUESSA_2
021,y_forecasted_BUEPOA_2021],axis=1)

df_final_TBATS.to_excel('TBATS_Proyecciones_2021.xlsx')

# Revisamos las proyecciones
print(df_final_TBATS)

# Guardamos los RMSE obtenidos en prueba para comparar los modelos

df_final_RMSE_TBATS=pd.concat([y_forecasted_BUESCL_2021,y_forecasted
_BUESAO_2021,y_forecasted_BUEMVD_2021,y_forecasted_BUESRZ_2021,y_for
ecasted_BUELIM_2021,y_forecasted_BUEASU_2021,y_forecasted_BUERIO_202
1,y_forecasted_BUEPDP_2021,y_forecasted_BUEFLN_2021,y_forecasted_BUE
SSA_2021,y_forecasted_BUEPOA_2021],axis=1)

df_final_RMSE_TBATS.to_excel('RMSE_TBATS_Proyecciones_2021.xlsx')

print(df_final_RMSE_TBATS)



### PREDICCIONES PARA 2022
# Dado que la red neuronal univariada fue la de mayor precisión en
los ejercicios de proyección para 2020 y 2021 se entrena dicha red
con todos los datos disponibles y se procede a realizar las
```

```
proyecciones a 12 meses en avance. Este proceso se realiza para las
11 rutas bajo análisis.

# 1) Preprocesamiento de los datos
# Cargamos nuevamente los datos originales

df_BUEXXX = df['BUEXXX'].values
df_BUEXXX = df_BUEXXX.reshape((-1,1))


# Filtramos los datos de entrenamiento, que van desde 2001 a 2021

split=252
train_BUEXXX = df_BUEXXX[:split]

print(len(train_BUEXXX))

# Definimos el tamaño de la secuencia con para definir las
secuencias de entrenamiento, utilizamos la función
TimeseriesGenerator para dicha tarea (look_back es lo mismo que
time_steps en la función create_dataset)

look_back = 12

train_generator_BUEXXX=TimeseriesGenerator(train_BUEXXX,train_BUEXXX
,length=look_back, batch_size = 42)

print(train_generator_BUEXXX)

# 2) Modelado

model = Sequential()
model.add(LSTM(100,input_shape=(look_back,1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error',
optimizer=keras.optimizers.Adam(0.01))

model.summary()

# 3) Entrenamiento del modelo
# Entrenamos el modelo con todos los datos y la misma
parametrización utilizada en la red neuronal univariada

model_BUEXXX = model.fit_generator(train_generator_BUEXXX,
epochs=100, batch_size= 42, shuffle = False, verbose=1)

# 4) Predicciones para 2022
# Para realizar las predicciones, se define la función predict, que
va incorporando dentro de las secuencias de input para las
predicciones del mes actual, las predicciones realizadas para el mes
anterior

df_BUEXXX = df['BUEXXX'].values
df_BUEXXX = df_BUEXXX.reshape((-1))

def predict(num_prediction, model_BUEXXX):
    prediction_list = df_BUEXXX[-look_back:]

    for _ in range(num_prediction):
        x = prediction_list[-look_back:]
        x = x.reshape((1, look_back, 1))
```

```
        out = model.predict(x)[0][0]
        prediction_list = np.append(prediction_list, out)
    prediction_list = prediction_list[look_back-1:]

    return prediction_list

# Definimos el horizonte de predicción y hacemos las predicciones

num_prediction = 12

forecast = predict(num_prediction, model_BUEXXX)

# Ajustamos el formato de las predicciones para luego guardarlas en
un archivo

forecast=forecast.reshape((13,))
forecast_BUEXXX=pd.Series(forecast)

print(forecast_BUEXXX)

# Luego de realizado el ejercicio de predicción para las 11 rutas
bajo análisis, los resultados obtenidos

from pandas import concat

df_final_predicciones_2022=pd.concat([forecast_BUESCL,forecast_BUESA
O,forecast_BUEMVD,forecast_BUESRZ,forecast_BUELIM,forecast_BUEASU,fo
recast_BUERIO,forecast_BUEPDP,forecast_BUEFLN,forecast_BUESSA,
forecast_BUEPOA],axis=1)

df_final_predicciones_2022.to_excel('LTMS_UNIV_Predicciones_2022.xls
x')

# Revisamos las predicciones
print(df_final_predicciones_2022)
```