

TÓPICOS METEOROLÓGICOS Y OCEANOGRÁFICOS

Ministerio de Ambiente y Energía - Instituto Meteorológico Nacional

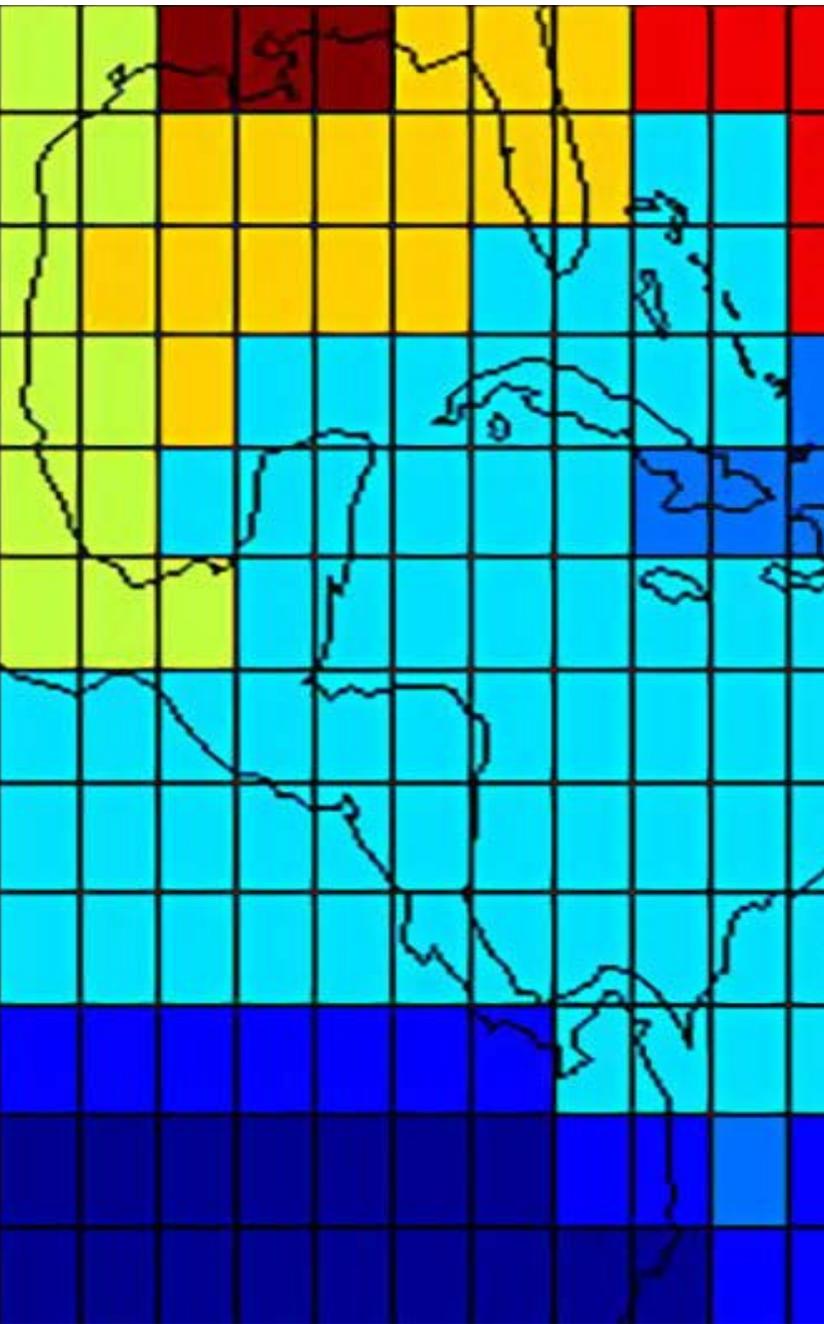


Instituto Meteorológico Nacional
Fundado en 1888

ISSN 2953-738X

Volumen 23

Diciembre 2024

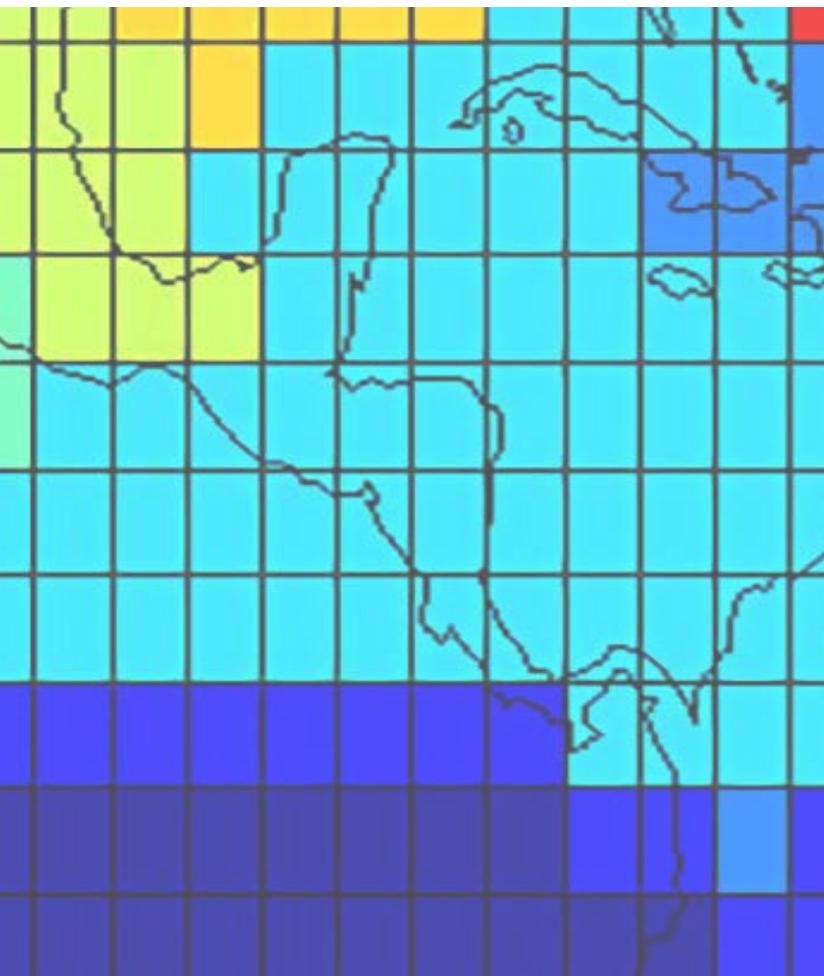


TÓPICOS METEOROLÓGICOS Y OCEANOGRÁFICOS

ISSN 2953-738X

Volumen 23

Diciembre 2024



Ministerio de Ambiente y Energía
Instituto Meteorológico Nacional

Coordinación general:
Mag. Eladio Solano León

Edición:
M.Sc. Gabriela Chinchilla Ramírez

Comité editorial:
M.Sc. Daniel Poleo Brito
M.Sc. Gabriela Chinchilla Ramírez
M.Sc. Karina Hernández Espinoza

Diseño y diagramación:
Rodrigo Granados Jiménez

Imágenes de portada:
Nury Sanabria Valverde

La revista "Tópicos Meteorológicos y Oceanográficos" es publicada por el Instituto Meteorológico Nacional, Ministerio de Ambiente y Energía, Costa Rica. Tiene como finalidad dar a conocer los resultados de investigaciones y estudios en Ciencias de la Atmósfera y Oceanografía Física.

Los artículos publicados en Tópicos Meteorológicos y Oceanográficos expresan la opinión del autor y no necesariamente del Instituto Meteorológico Nacional

CONTENIDO

Pág. Artículo

5 Interacciones océano-atmósfera en la Piscina de Agua Cálida del Hemisferio Occidental

FRANCISCO JOSÉ GAVIDIA MEDINA

15 Python aplicado a la meteorología operativa: ejemplo práctico para la meteorología aeronáutica

ROBERTO CAMILO VINDAS MORÁN
DIEGO GARRO MOLINA

Python aplicado a la meteorología operativa: ejemplo práctico para la meteorología aeronáutica

ROBERTO CAMILO VINDAS MORÁN¹
DIEGO GARRO MOLINA²

Resumen

El uso de lenguajes de programación, y en específico Python, está muy extendido entre los meteorólogos en la academia e investigación, sin embargo, no suele ser el caso en la meteorología operativa. Muchas de las tareas que los meteorólogos operativos deben realizar por su naturaleza pueden ser tediosas o lentas, pero en muchos casos presentan características que permiten mejorarlas mediante el uso de software específico. En este artículo los autores buscan estimular a más meteorólogos operativos a aprender y aplicar Python para la optimización de las tareas diarias que deben realizar, así como fomentar una mayor incorporación de la meteorología operativa a las comunidades de intercambio y colaboración de software científico. Se argumentan las ventajas de Python en el proceso de aprendizaje incluso en condiciones autodidactas, además se introduce brevemente las herramientas GIT y GITHUB para el intercambio y colaboración en el desarrollo de software. Con este fin se presentan tres ejemplos aplicados de Python específicamente en meteorología aeronáutica para la automatización de descarga y almacenamiento de los datos de metares, la elaboración de pronósticos de aeródromo y elaboración de las carpetas de despegue.

PALABRAS CLAVE: METEOROLOGÍA OPERATIVA, METEOROLOGÍA AERONÁUTICA, PYTHON, CONTROL DE CALIDAD, PRONÓSTICO DEL TIEMPO, PROGRAMACIÓN CIENTÍFICA, METAR, TAF, GIT

Abstract

The use of programming languages, and specifically Python, is widespread among meteorologists in academia and research, however, this is not usually the case in operational meteorology. Many of the tasks that operational meteorologists must perform by their nature can be tedious or time-consuming, but in many cases they have features that allow them to be improved through the use of specific software. In this article the authors seek to encourage more operational meteorologists to learn and apply Python for the optimization of the daily tasks they must perform, as well as to encourage greater incorporation of operational meteorology into scientific software sharing and collaboration communities. The advantages of Python in the learning process even under self-taught conditions are argued, furthermore, the GIT and GITHUB tools for software development sharing and collaboration are briefly introduced. To this end, three applied examples of Python are presented, specifically in aeronautical meteorology for the automation of download and storage of metars data, the elaboration of aerodrome forecasts and the elaboration of take-off forecasts.

KEYWORDS: OPERATIONAL METEOROLOGY, AERONAUTICAL METEOROLOGY, PYTHON, QUALITY CONTROL, WEATHER FORECASTING, SCIENTIFIC PROGRAMMING, METAR, TAF, GIT

-
- 1 Bachiller en Física y Meteorología. Departamento de Meteorología Sinóptica y Aeronáutica, Instituto Meteorológico Nacional (IMN). Correo electrónico: rvindas@imn.ac.cr.
 - 2 Bachiller en Meteorología. Departamento de Meteorología Sinóptica y Aeronáutica, Instituto Meteorológico Nacional (IMN). Correo electrónico: dgarro@imn.ac.cr.

1. INTRODUCCIÓN

La evolución de las ciencias de la computación ha ido siempre en conjunto con el de las ciencias básicas, las cuales han utilizado con gran entusiasmo sus herramientas en sus investigaciones y han aportado también a dicha evolución. En las últimas décadas el progreso de la tecnología ha hecho herramientas como los lenguajes de programación más accesibles y ha convertido su manejo en una necesidad para prácticamente todas las áreas de la ciencia.

En este aspecto destaca el lenguaje de programación Python el cual es bastante sencillo de aprender principalmente porque su sintaxis se acerca mucho al lenguaje humano, es multiplataforma lo cual implica que puede ser utilizado en varios sistemas operativos y cuenta con un gran número de librerías específicas para aplicaciones científicas (Ayer et. Al, 2014).

Otro elemento importante a resaltar de Python es la existencia de enormes comunidades de soporte y colaboración en línea lo que facilita la consulta y resolución de problemas. Esta característica es especialmente útil en situaciones en donde se debe realizar un esfuerzo autodidacta para realizar un determinado proyecto. Un ejemplo de estas plataformas es el sitio stack overflow (<https://stackoverflow.com/>) el cual tiene el formato de pregunta-respuesta (Q/A) y es uno de los más utilizados como recurso para mejorar o aprender a programar (Stack Overflow, 2024).

En años recientes la utilización de Python en la meteorología ha ido en aumento y múltiples paquetes de librerías específicamente para aplicaciones meteorológicas han sido desarrolladas en las últimas décadas. Entre los ejemplos más exitosos se pueden mencionar los casos de MetPY (May et. al. 2022) y MetSim (Bennette et. al, 2020), siendo la primera una herramienta para el análisis y visualización de datos y la segunda para procesar los datos de

ingesta de modelos numéricos meteorológicos con las correctas estimaciones y rangos temporales.

Estas características convierten a Python en un lenguaje muy poderoso para fines de meteorología aeronáutica operativa, especialmente porque permite a los meteorólogos desarrollar sus propias herramientas para resolver problemas específicos y facilita a los servicios meteorológicos generar su propia innovación sin necesidad de adquirir software por vía privada lo que significa un ahorro

Por otro lado, desde el nacimiento de la aviación fue evidente que las condiciones meteorológicas tienen una afectación directa en el desempeño exitoso de la misma, no sólo por motivos de seguridad y resguardo de la vida, sino también por motivos económicos y de gestión. De esta manera el desarrollo de la aviación ha ido de la mano con el de la meteorología aeronáutica mediante el estudio de las variables meteorológicas y su impacto en todas etapas del vuelo (Méndez & Alcázar, 2018).

En las últimas décadas se han presentado avances en ingeniería aeronáutica que han permitido construir aeronaves más eficientes y con mejor respuesta ante inclemencias del tiempo. Esto sumado a las mejorías en el pronóstico y monitoreo del tiempo han disminuido la siniestralidad aeronáutica por causas meteorológicas en el mundo.

Aun así, los efectos del tiempo severo continúan representando un riesgo de seguridad. (Moreno & Gil 2003) analizaron la responsabilidad de las condiciones meteorológicas en los accidentes aéreos a escala global entre 1970-1999 y encontraron que la cantidad de accidentes aéreos por esta causa representaba una parte pequeña del total (9.5% en los años 70 y 3.9% en los 90), siendo el engelamiento y la cizalladura los fenómenos más peligrosos. En otro estudio más reciente, (Bezek et al. 2023) realizaron un trabajo similar pero acotado a 8 países de

Sudamérica y 1 de Centroamérica encontrando una responsabilidad del 10.4 % de factores meteorológicos en accidentes aéreos para el periodo de 2013-2022.

Otro elemento muy importante que ha sido indispensable en la disminución en los incidentes aéreos es la estandarización y regulación de los procedimientos en materia aeronáutica en general, incluyendo la información meteorológica. Actualmente los entes principales que regulan estas normas son la Organización Meteorológica Mundial (OMM) y la Organización Aeronáutica Civil Internacional (OACI). En la parte meteorológica estos organismos determinan el tipo de variable que se mide, la manera en que se debe reportar y almacenar, así como los controles de calidad necesarios. También establecen directrices en el procedimiento y métodos de capacitación del personal a cargo de estas labores (Valverde, 2017).

El rápido crecimiento de la industria aeronáutica en los últimos años previos a la pandemia del COVID 19 había traído el desafío de mejorar las técnicas y tecnologías empleadas en materia de meteorología aeronáutica ante el incremento del tráfico aéreo. Ahora en época postpandemia en el que la industria se recupera progresivamente (España, 2022) esa necesidad de mejora debería de estar nuevamente en la palestra.

Dentro de este marco se encuentran esfuerzos de investigación como los de (Marvé et. al ,2021) quienes destacan la necesidad de automatización de productos específicos de información meteorológica para que estos pueden estar a la par con el desarrollo de tecnologías e innovación en procedimientos de navegación aérea, como es el caso por ejemplo de las llegadas basadas en tiempo (Time Based Separation) que aumentan la capacidad del aeropuerto y requieren de información meteorológica precisa en tiempo real.

Esta actualización y mejora de procesos en la meteorología aeronáutica se puede conseguir

mediante la creación de software especializado. Es en este marco que el Instituto Meteorológico Nacional (IMN) desarrolló los programas presentados en este trabajo y que se describen en las secciones siguientes. Es importante destacar que los realizadores de los programas que se presentan no son profesionales en computación o ingeniería de software, sino meteorólogos que utilizaron metodologías autodidactas y diferentes recursos en línea para elaborar los programas.

2. METODOLOGÍA

A continuación, se describe los programas realizados, los datos utilizados y las herramientas para la elaboración de los mismos. En el cuadro 1 se resumen algunos detalles como la licencia de distribución, el nombre de los programas, las versiones de Python en la que han sido probados exitosamente y los enlaces para el acceso al código fuente de los mismos.

El acceso al código fuente se brinda utilizando el sistema de control de versiones GIT mediante la plataforma en línea github (<https://github.com>). El control de versiones es un sistema que registra los cambios en un archivo o conjunto de archivos a lo largo del tiempo para que pueda recuperar versiones específicas más adelante (Chacon & Straub 2014). La ventaja de este tipo de sistemas en la elaboración de software, especialmente si hay más de una persona trabajando en el proyecto, es que permite revertir archivos de scripts o incluso el proyecto entero a estados anteriores, comparar cambios en el tiempo, identificar causantes de problemas en el código entre otras funciones.

En la figura 1 se muestra una captura de pantalla para el primer enlace del cuadro 1 (crea Carpetas 2.5). Los pasos para descargar el código se señalizan con flechas rojas.

Un procedimiento similar se utiliza con los demás enlaces del cuadro 1. No es necesario tener

instalado GIT para acceder al código fuente mediante la página de github. Sin embargo, si se tiene instalado GIT en la computadora local,

se puede clonar el repositorio de los proyectos utilizando la terminal de GIT mediante el comando git clone y los enlaces del cuadro 1. El

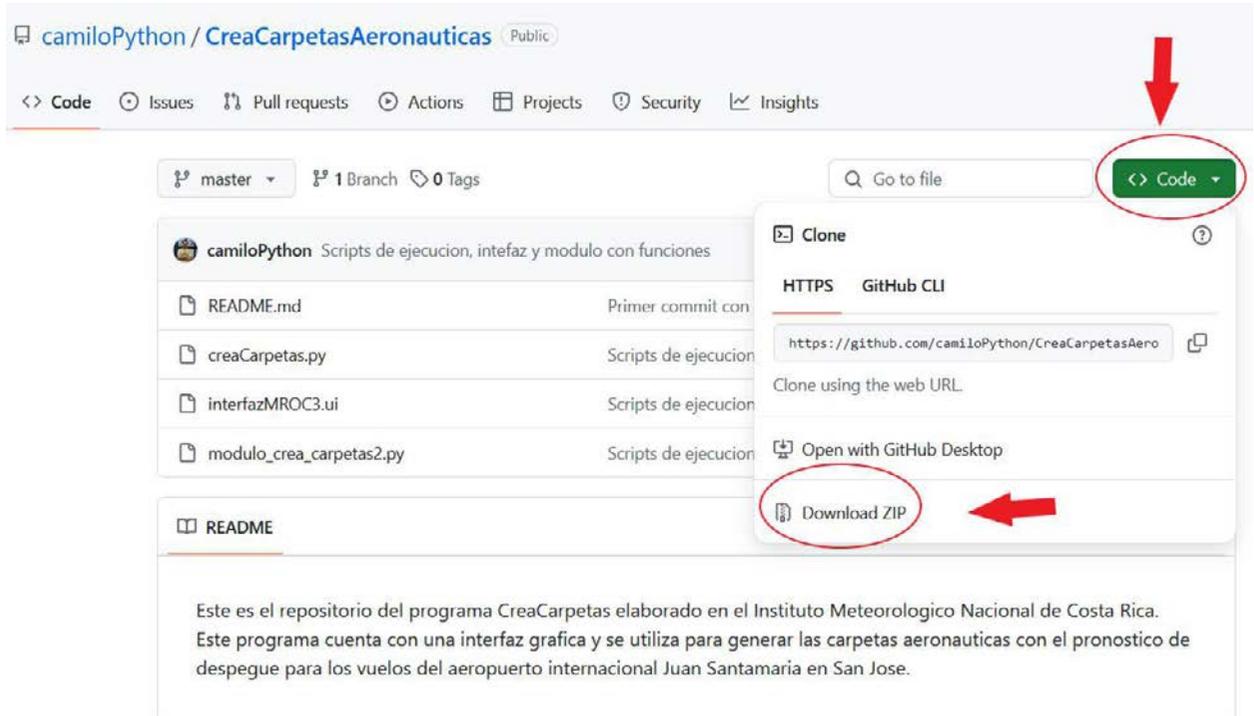


Figura 1. Captura de pantalla de la página github.com para el proyecto de Crea Carpetas 2.5. Los pasos para descargar el código se marcan con flechas y círculos rojos.

Cuadro 1. Especificaciones y acceso al código fuente de los programas

Programa	Licencia	Versiones de Python en que ha sido probado	Librerías principales	Código fuente
Crea Carpetas 2.5	Boost Software License 1.0	Python 3.9.13	Reportlab 3.5.67 PyQT5 5.15.7 Pillow 10.2.0	https://github.com/camiloPython/CreaCarpetasAeronauticas
get-metars 0.5.2	MIT	Python >=3.8	requests 2.27.1 beautifulsoup 4.11.1 aiohttp 3.8.1 html5lib 1.1 pydantic 1.9.1 typer 0.9.0	https://github.com/TAF-Verification/get-metars
metar-forecast 0.0.1	MIT	Python >=3.8	get-metars 0.3.1 aeromet-py 0.1.2 pandas 2.0.1 aiohttp 3.8.4 sternum 0.4.10 rocketry 2.5.1 fastapi 0.1 uvicorn 0.22 jinja2 3.1.2	https://github.com/diego-garro/metar-forecast

lector interesado que quisiera aprender más de GIT puede encontrar muchos recursos en línea, además de la documentación oficial (Chacon & Straub 2014).

2.1. Crea Carpetas 2.5

El primer ejemplo de Python aplicado a la meteorología aeronáutica operativa que se presenta es el programa Crea Carpetas 2.5. La función de este programa es generar las carpetas de pronóstico de despegue para los vuelos en el aeropuerto internacional Juan Santamaría (AIJS) en San José, Costa Rica.

De acuerdo con la OMM dentro de los procedimientos de las oficinas meteorológicas en la aviación se encuentra la elaboración de los pronósticos de despegue los cuales se entregan a los operadores y miembros de la tripulación alrededor de unas tres horas antes del vuelo (OMM, 2003). El IMN con su oficina en el AIJS realiza esta

labor mediante la confección de carpetas de vuelo aeronáuticas, las cuales son documentos que se envían por correo electrónico con previa solicitud de los despachadores de vuelo de cada aerolínea. Alternativamente existe la opción de entregar las carpetas impresas directamente a los pilotos si estos así lo solicitan. En la figura 2 se muestra un ejemplo de una se estas carpetas.

Estos documentos cuentan con una portada donde se incluye los detalles del vuelo como su código, hora de salida, la fecha, el nombre del pronosticador y el pronóstico para la hora del despegue de las variables meteorológicas de interés a nivel del aeródromo, las cuales son: temperatura, dirección e intensidad del viento y presión atmosférica.

En las siguientes páginas se incluye información del pronóstico de aeródromo (*Terminal Aerodrome Forecast, TAF*) de los aeropuertos de escala y destino final que se toman de la página web del Aviation Weather Center (<https://aviationweather>.

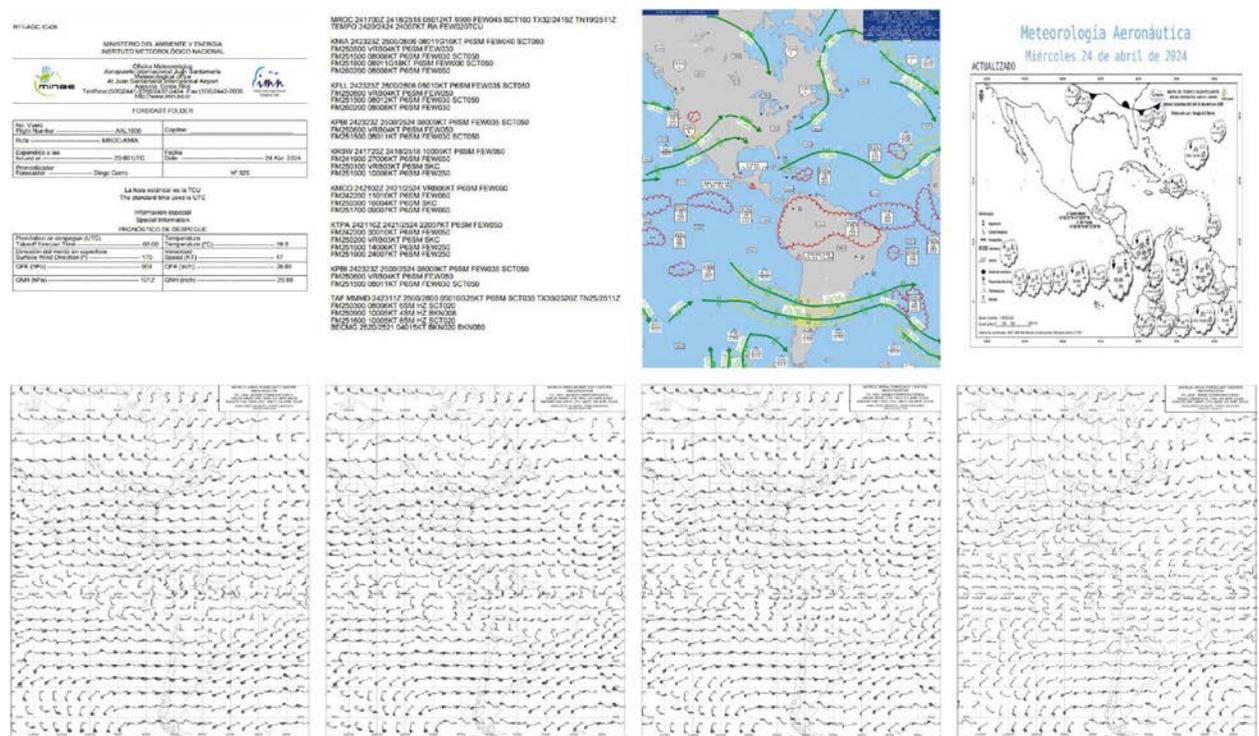


Figura 2. Páginas de una carpeta aeronáutica de vuelo elaborada en el AIJS.

gov), así como de los aeropuertos alternos para el vuelo específico. Se incluye también mapas de tiempo significativo, uno elaborado por el World Area Forecast Center (WAFC) del Met Office del Reino Unido y otro elaborado por la oficina meteorológica del Aeropuerto Internacional Tobías Bolaños Palma de Costa Rica. Finalmente se incluyen mapas de viento y temperatura del WAFC de Washington para cuatro distintos periodos de pronóstico (+6 horas, +12 horas, +18 horas y +24 horas) que dependen de la hora del vuelo en específico.

Anteriormente los pronosticadores del AIJS realizaban estas carpetas mediante un método tradicional, haciendo una a la vez, elaborando el pronóstico de despegue, descargando e insertando cada imagen necesaria, además de copiar y pegar uno por uno los datos de los TAF de cada aeropuerto. De acuerdo a consultas realizadas a los pronosticadores, dependiendo de la cantidad de vuelos y a las condiciones meteorológicas presentes en cada turno respectivo, esta actividad les podía tomar desde una hora hasta un máximo de dos horas con treinta minutos. La elaboración de las carpetas se realiza durante los tres turnos de pronóstico en la oficina meteorológica del aeropuerto. El primer turno se realiza de 6am a 2pm, el segundo de 2pm a 9pm y el tercero de 9pm a 6am del día siguiente.

Ante la necesidad de optimizar esta labor se estableció en el 2016 el proyecto de crear una herramienta digital de fácil uso e intuitiva que mejorara el proceso de creación de estas carpetas aeronáuticas.

Para realizar este proyecto se decidió utilizar el lenguaje Python debido a su relativa sencillez y principalmente al enorme soporte y comunidades en línea que existen para este. Se utilizó el paquete de distribución de software Anaconda (Anaconda Software Distribution, 2020) para la instalación de las librerías necesarias, así como el *entorno integrado de desarrollo* (IDE, por sus siglas en

inglés) Spyder (Raybaut, 2009) para escribir el código del programa.

El código fue realizado originalmente en la versión 2.7 de Python, sin embargo, este se migró a Python 3 en el 2020 debido a que Python 2.7 dejó de tener soporte ese año.

Para abordar el problema lo primero que se determinó fueron las características que las carpetas comparten entre si, con el fin de utilizar estos factores de manera ventajosa, por ejemplo, en la elaboración de varias carpetas en paralelo.

Al realizar este procedimiento se observó que, debido al amplio rango de validez de los mapas de tiempo significativo y de temperatura y viento (de 6 hasta 24 horas), muchas de las carpetas de vuelos distintos compartían los mismos mapas lo que implica que sólo se necesitan descargar los mapas una vez por turno.

También se observó que para vuelos con horas muy cercanas de despegue las variables pronosticadas son en su mayoría muy similares, de manera que la diferencia más significativa está en los TAF asociados a ese vuelo particular.

Para procesar esta información de manera más eficiente se generó un archivo de texto separado por comas con la información del código del vuelo y los códigos de todos los aeropuertos asociados a dicho vuelo (figura 3).

Por otro lado, se observó que los valores de presión (QFE y QNH) y su conversión de hPa a pulgadas de mercurio (inch) está determinado por un solo valor de pronóstico ya que se cuenta con una tabla con los valores asociados. Esta información se recopiló en otro archivo de texto separado por espacios (figura 3).

Para facilitar la capacitación y uso del programa se decidió que este debía contar con una interfaz gráfica la cual se muestra en la figura 4 y fue

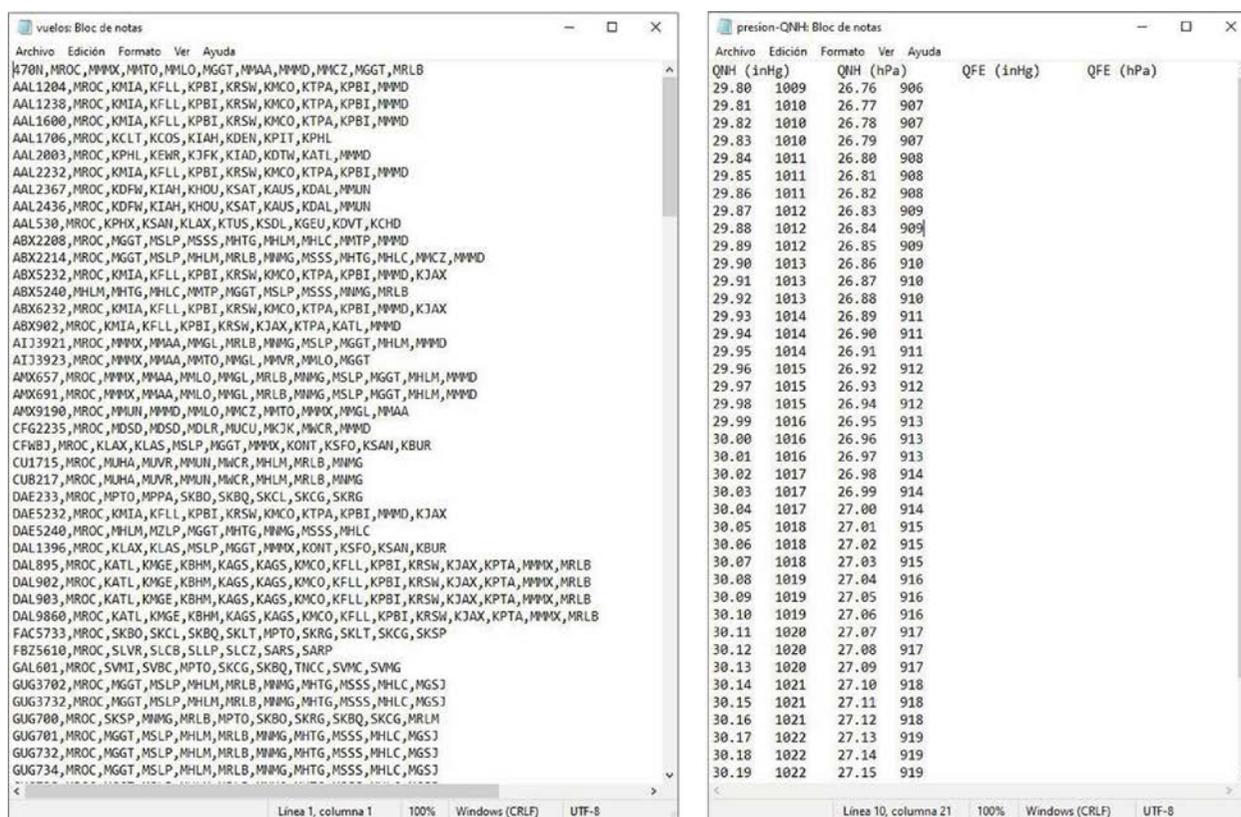


Figura 3. Ejemplos de archivos de texto con la información de los vuelos (izquierda) y los valores de presión atmosférica (derecha).

diseñada usando la herramienta QT Designer (<https://doc.qt.io/qt-6/qtdesigner-manual.html>). Los elementos de la interfaz se describen a continuación.

- 1) Lista de selección en donde se selecciona el aeropuerto nacional en donde se está realizando la carpeta (Alajuela, Pavas, Liberia o Limón).
- 2) En esta lista de selección se despliegan los códigos de vuelo que se toma del primer elemento de cada línea (470N, AAL1204, etc...) en el archivo de la izquierda en la figura 2. Para agregar vuelos nuevos o quitar alguno de los ya existentes se modifica el archivo de texto agregándole o quitándole líneas nuevas siguiendo el mismo formato (Se debe volver a iniciar el programa para observar los cambios).
- 3) En este cuadro de texto el pronosticador digita su nombre usando el teclado.
- 4) En este cuadro de texto el pronosticador digita el número de carpeta según el registro que llevan en el día.
- 5) En este cuadro de texto se digita la hora de emisión de la carpeta.
- 6) En este cuadro de texto se digita la hora del pronóstico de despegue.
- 7) En esta lista de selección se despliegan los valores de la primera columna del archivo de la derecha en la figura 2. El pronosticador elige uno y los demás se escriben en la carpeta por defecto.
- 8) En este cuadro de texto se digita la dirección del viento pronosticada en superficie.
- 9) Se digita la velocidad del viento en nudos.
- 10) Se digita la temperatura pronosticada en grados Celsius.
- 11) Caja de texto para observaciones adicionales.
- 12) Casillas en donde el pronosticador selecciona la validez de los mapas de tiempo significante.

Figura 4. Interfaz gráfica de Crea Carpetas 2.5.

- 13) Check Box en donde el pronosticador selecciona la validez de los mapas de viento y temperatura.
- 14) Check Box en donde el pronosticador selecciona si se trata de un vuelo alto (opción por defecto) o un vuelo bajo (suelen ser vuelos locales).

- 15) En caso de seleccionar vuelo bajo en el punto anterior se da la opción de agregar mapas elaborados por el mismo pronosticador utilizando un programa de terceros llamado Zygrib (disponible en <https://www.zygrib.org/>). Sin embargo, este programa genera los mapas a colores y no se ve bien a la hora

de imprimirlos por lo que estos controles giratorios dan la opción de aplicar un factor de gris para aclarar u oscurecer la imagen y que se vea mejor impresa. El valor del factor (rango de 1 a 100) es decidido por el pronosticador.

16) Función para descargar los mapas. Esta es la función más importante y la que marca mayor diferencia al momento de optimizar la elaboración de las carpetas. Este botón sólo es necesario de aplicar una vez por turno ya que descarga **todos** los mapas indistintamente si se van a usar o no y permite que el pronosticador solo deba elegir mediante las diferentes opciones que mapa agregar en cada carpeta. Los mapas se descargan utilizando la hora y fecha del servidor local como referencia.

17) Función para crear la carpeta de vuelo.

En la figura 5 se muestra un esquema con una secuencia de pasos para utilizar el programa. Esta secuencia no es estricta ya que los pasos entre a) y f) pueden ejecutarse en cualquier orden sin alterar el desempeño del programa. El único paso que se ejecuta en un orden específico es el g) que se realiza de último cuando se crea la carpeta.

En cada turno de pronóstico la primera carpeta se puede realizar siguiendo el esquema entre los pasos a) hasta g). Sin embargo, para las siguientes carpetas sólo es necesario alternar entre los pasos b), e) y g) sin necesidad de pasar por ninguno de los demás. Eventualmente podría ser del interés del pronosticador seleccionar un mapa con periodo de validez distinto en los pasos c) o d) pero no es necesario volver a descargarlos, es decir, el paso a) solo se ejecuta una vez por turno. Luego de ejecutar el paso g) el programa genera la carpeta en formato WORD con todos los mapas incluidos. Esta característica es la principal ventaja del programa y lo que permite realizar las carpetas mucho más rápido en comparación al método tradicional.

Para implementar el programa se generó un archivo ejecutable utilizando la librería pyinstaller

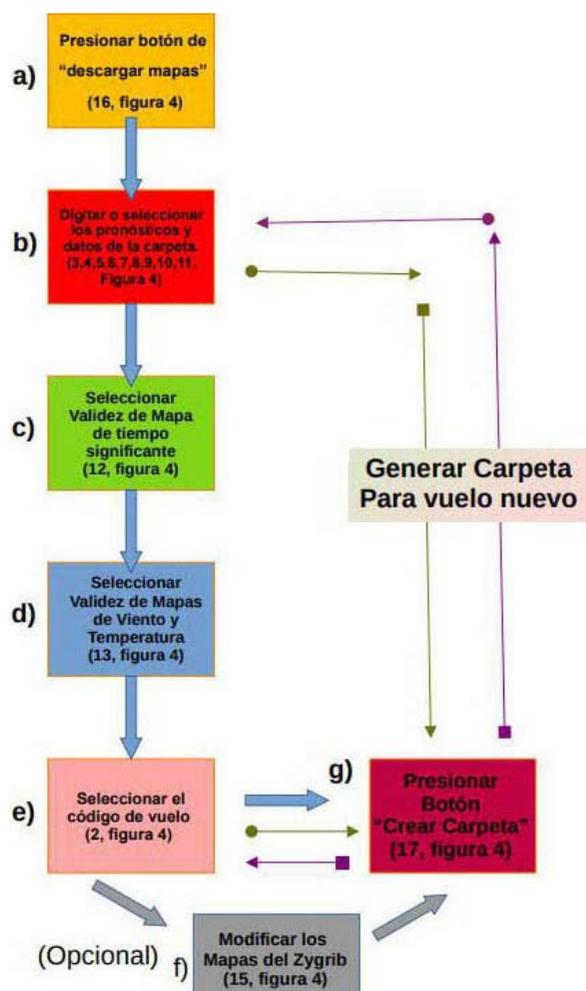


Figura 5. Diagrama de flujo para la utilización del programa.

(<https://pyinstaller.org/en/stable/>) creando así una versión portable que se puede utilizar en cualquier computadora con Windows 10 u 11 sin necesidad de tener python instalado.

2.2. get-metars 0.5.2

Las Oficinas de Meteorología Aeronáutica (OMA's) destacadas en los diferentes aeropuertos alrededor del mundo generan productos y servicios utilizados diariamente por los usuarios de dichas terminales aeroportuarias. Tales productos y servicios se resumen en mensajes codificados y se almacenan en archivos de

Microsoft Excel y Microsoft Word según las reglas de la Organización de Aviación Civil Internacional (2018) y la Organización Meteorológica Mundial (2018). De este modo, los datos de un día natural son almacenados en un archivo independiente, por lo que al cabo un año se cuenta con 365 archivos para cada tipo de mensaje elaborado por la OMA, uno para cada día del año.

Los METAR (Météorologique Aviation Régulière por sus siglas en francés) y TAF (Terminal Aerodrome Forecast por sus siglas en inglés) (Organización de Aviación Civil Internacional, 2018) son los reportes meteorológicos de aeródromo y son almacenados en archivos digitales tipo Microsoft Excel y Microsoft Word diariamente. Este formato, a pesar de estar en regla con las entidades oficiales antes mencionadas, representa una dificultad para la manipulación a gran escala de los datos ahí almacenados por varias razones que se enumeran a continuación:

- 1) La gran cantidad de archivos.
- 2) Errores humanos de digitación de los datos almacenados.
- 3) Celdas combinadas dentro del documento.
- 4) Cambios de versión, edición o formato del archivo a lo largo del tiempo.

Aunque es factible lograr cierto grado de automatización con este formato de almacenamiento de datos, los mencionados inconvenientes auguran un panorama desafiante en el que persiste la necesidad de mucho trabajo manual para desarrollar nuevos productos. Por otro lado, los reportes meteorológicos METAR también son transmitidos por las OMA's de todo el mundo a los nodos de recopilación de datos internacionales en internet, lo que facilita el acceso a los mismos a través de sitios web específicos. Un ejemplo es el sitio Ogimet.com, que almacena datos meteorológicos aeronáuticos desde el año 2005 y es utilizado por `get-metars 0.5.2` para obtener y almacenar la información necesaria.

Este tipo de sitios web representan una oportunidad de acceso a una mayor cantidad de datos de manera más expedita, concisa y sobre todo sencilla, si hablamos de automatizar tareas. Con esto surge la idea de la creación del paquete `get-metars 0.5.2`, el cuál es una aplicación de línea de comandos (CLI por sus siglas en inglés) el cual permite descargar los METAR y TAF desde la terminal de cualquier computadora conectada a internet, y sin tener que recurrir al navegador web. Cuenta con documentación sencilla y concisa como se muestra en la figura 6. `get-metars` está escrito con el lenguaje de programación Python y se encuentra alojado en el nodo de recolección de paquetes oficial de Python, PiPy.org. De esta manera, `get-metars` puede instalarse utilizando el procedimiento regular de instalación de paquetes oficial de Python, el módulo "pip" con el comando: `pip install get-metars`.

2.3. metar-forecast 0.0.1

Como parte de las labores realizadas en la OMA del Aeropuerto Internacional Juan Santamaría, surge la necesidad de contar con datos pronosticados de QNH (presión al nivel de la estación), temperatura, dirección y velocidad del viento. Dichos datos son solicitados tanto por los despachadores de vuelo vía telefónica como por los vuelos internacionales adjuntados en las carpetas aeronáuticas elaboradas con la aplicación antes mencionada, *Crea Carpetas 2.5*. Se creó entonces, una aplicación web llamada *metar-forecast*, con el fin de suplir dichas necesidades de manera constante.

2.3.1 Datos

La aplicación *metar-forecast* calcula un valor medio de las variables de interés, usando los datos de los METAR de la estación en cuestión: dirección, velocidad y ráfagas del viento, QNH, temperatura del aire y temperatura de rocío.

```

> get-metars --help
Usage: get-metars [OPTIONS] [ICAO]

Arguments:
  [ICAO] The ICAO code of the station to request, e.g. MROC for Int. Airp.
         Juan Santamaría [default: MROC]

Options:
  -t, --type [SA|SP|FT|FC|ALL]  Type of report to request. SA -> METAR, SP
                                -> SPECI, FT -> TAF (long), FC -> TAF
                                (short), ALL -> All types [default:
                                ReportType.SA]
  -i, --init [%Y-%m-%d|%Y-%m-%dT%H:%M:%S|%Y-%m-%d %H:%M:%S]
                                The initial UTC date and time to request the
                                reports. [default: 2006-01-01T00:00:00]
  -f, --final [%Y-%m-%d|%Y-%m-%dT%H:%M:%S|%Y-%m-%d %H:%M:%S]
                                The final UTC date and time to request the
                                reports. Defaults to `init` + 30 days, 23
                                hours and 59 minutes.
  -F, --file TEXT
                                The filename to write the reports on disk.
                                Default will be changed [default:
                                metar.txt]
  -o, --one-line
                                Removes white spaces in the reports. If True
                                reports will be written in one line.
  -s, --sanitize
                                Sanitizes the report to use in TAF
                                verification program.
  --old-first / --no-old-first
                                Writes the reports ordered by date older
                                first. If no, writes the reports newer
                                first. [default: old-first]
  --datetime-prefix / --no-datetime-prefix
                                Adds the date and time as a prefix of the
                                reports with format `%Y%m%d%H%M` [default:
                                datetime-prefix]
  --install-completion [bash|zsh|fish|powershell|pwsh]
                                Install completion for the specified shell.
  --show-completion [bash|zsh|fish|powershell|pwsh]
                                Show completion for the specified shell, to
                                copy it or customize the installation.
  --help
                                Show this message and exit.

```

Figura 6. Opciones y funciones que puede realizar get-metars 0.5.2. Esta información se obtiene al ejecutar: \$ get-metars --help.

Se realiza el mismo proceso para los cuatro aeropuertos internacionales de Costa Rica, los cuales se listan en el cuadro 2.

Con el uso de get-metars 0.5.2 se descargan los METAR de cada una de las estaciones, aplicación que los almacena en un archivo de texto. Como el sitio web de Ogimet.com cuenta con datos históricos desde el año 2005, se crea una script de Bash con el cual se pueden descargar todos los datos de un año entero en alrededor de 6 minutos, este script se presenta en la figura 7. Se realiza una pausa de 20 segundos entre cada petición ya que,

Cuadro 2. Aeropuertos internacionales de Costa Rica y su código ICAO (Organización de Aviación Civil Internacional, 2020).

Estación (Aeropuerto Internacional)	Código ICAO
Juan Santamaría	MROC
Tobías Bolaños Palma	MRPV
Daniel Oduber Quirós	MRLB
Limón	MRLM

Ogimet.com cuenta con un sistema de bloqueo de URL en caso que se realicen muchas peticiones en periodos cortos de tiempo.

Cuadro 3. Umbrales definidos para las variables de interés.

Variable	Umbral
Dirección del viento (°)	±20.0°
Velocidad del viento (kt)	±4.0 kt
Ráfagas de viento (kt)	±5.0 kt
Temperatura (°C)	±1.0°C
Temperatura de punto de rocío (°C)	±1.0°C
Presión QNH (inHg)	±0.02 inHg

Una vez descargados los datos, se obtienen los archivos de texto, cada archivo contiene los METAR de todo un año (ver Figura 8). Posterior a ello se extraen los datos de las variables meteorológicas de interés (dirección, velocidad y ráfagas de viento, temperatura y temperatura del punto de rocío, presión (QNH) en la estación) de los METAR, para lo cual se crea un script y se usa el paquete aeromet-py 0.1.2 el cual identifica

```

YEAR=2022
STATION=mrpv

download:
# January
get-metars -i $(YEAR)-01-01 -f $(YEAR)-01-31T23:59:00 -o $(STATION)
cat metar.txt > $(STATION)/$(YEAR).txt
sleep 20
# February
get-metars -i $(YEAR)-02-01 -f $(YEAR)-02-28T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# March
get-metars -i $(YEAR)-03-01 -f $(YEAR)-03-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# April
get-metars -i $(YEAR)-04-01 -f $(YEAR)-04-30T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# May
get-metars -i $(YEAR)-05-01 -f $(YEAR)-05-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# June
get-metars -i $(YEAR)-06-01 -f $(YEAR)-06-30T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# July
get-metars -i $(YEAR)-07-01 -f $(YEAR)-07-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# August
get-metars -i $(YEAR)-08-01 -f $(YEAR)-08-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# September
get-metars -i $(YEAR)-09-01 -f $(YEAR)-09-30T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# October
get-metars -i $(YEAR)-10-01 -f $(YEAR)-10-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# November
get-metars -i $(YEAR)-11-01 -f $(YEAR)-11-30T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
sleep 20
# December
get-metars -i $(YEAR)-12-01 -f $(YEAR)-12-31T23:59:00 -o $(STATION)
cat metar.txt >> $(STATION)/$(YEAR).txt
rm metars.txt

```

Figura 7. Script de Bash para descargar los datos de un año entero desde Ogimet.com, usando la aplicación get-metars 0.5.2.

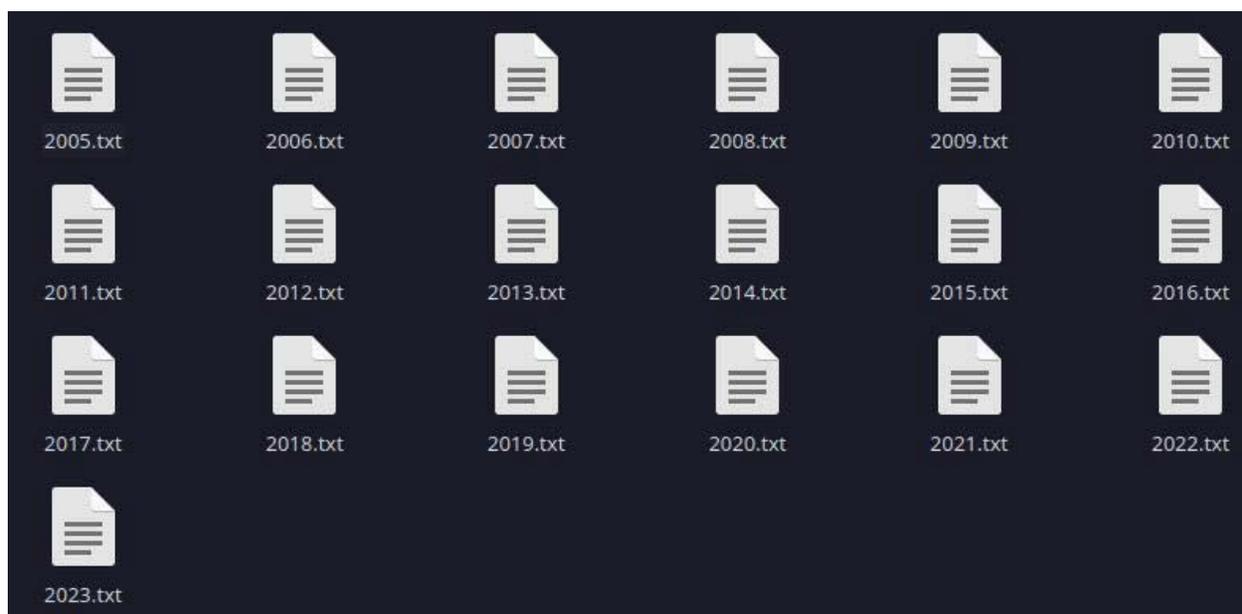


Figura 8. Vista de los archivos de texto que contienen los METAR, un archivo por cada año.

 A screenshot of a CSV file named 'metars.csv' in a spreadsheet application. The first row is the header, listing 16 columns: Year, Month, Day, Hour, Minute, Station, Wind_direction, Wind_speed, Wind_gust, Visibility, Cavok, Weather_intensity, Weather_description, Weat, and another column. The following rows contain numerical data for the year 2005, with values for each column separated by commas. Some cells contain 'NaN' or '0'.

Year	Month	Day	Hour	Minute	Station	Wind_direction	Wind_speed	Wind_gust	Visibility	Cavok	Weather_intensity	Weather_description	Weat	
2005	1	1	0	0	MROC	110	0	11	0	NaN	10000	0	0	NaN
2005	1	1	1	0	MROC	100	0	7	0	NaN	10000	0	0	NaN
2005	1	1	2	0	MROC	NaN	3	0	NaN	10000	0	0	NaN	NaN
2005	1	1	3	0	MROC	60	0	8	0	NaN	10000	0	0	NaN
2005	1	1	4	0	MROC	80	0	4	0	NaN	10000	0	0	NaN
2005	1	1	5	0	MROC	70	0	3	0	NaN	10000	0	0	NaN
2005	1	1	6	0	MROC	110	0	14	0	NaN	10000	0	1	NaN
2005	1	1	10	0	MROC	60	0	6	0	NaN	10000	0	1	NaN
2005	1	1	11	0	MROC	80	0	3	0	NaN	10000	0	1	NaN
2005	1	1	12	0	MROC	30	0	0	0	NaN	10000	0	0	NaN
2005	1	1	13	0	MROC	110	0	11	0	NaN	10000	0	0	NaN
2005	1	1	14	0	MROC	110	0	14	0	NaN	10000	0	0	NaN
2005	1	1	15	0	MROC	110	0	15	0	NaN	10000	0	0	NaN
2005	1	1	16	0	MROC	110	0	15	0	NaN	10000	0	0	NaN
2005	1	1	17	0	MROC	100	0	14	0	NaN	10000	0	0	NaN
2005	1	1	18	0	MROC	80	0	12	0	NaN	10000	0	0	NaN
2005	1	1	19	0	MROC	100	0	12	0	NaN	10000	0	0	NaN
2005	1	1	20	0	MROC	90	0	16	0	NaN	10000	0	0	NaN
2005	1	1	21	0	MROC	80	0	16	0	NaN	10000	0	0	NaN
2005	1	1	22	0	MROC	90	0	16	0	NaN	10000	0	0	NaN
2005	1	1	23	0	MROC	80	0	14	0	NaN	10000	0	0	NaN

Figura 9. Parte del archivo CSV generado a partir de los METAR, cada fila corresponde a los datos extraídos de un METAR.

los caracteres correspondientes a las variables de interés dentro del cuerpo del METAR. Con los datos identificados, se crea posteriormente un archivo CSV (Mitlöhner et al., 2016) organizado por columnas, parte de dicho archivo se presenta en la figura 9. Cabe destacar que estos datos fueron revisados y corregidos manualmente dentro de los archivos de texto anuales, de modo que se minimicen los errores de identificación

por valores mal digitados durante el proceso de confección de dichos mensajes en las OMA's.

2.3.2 Descripción del modelo

Debido a la naturaleza de los datos, se realiza un proceso empírico para la obtención de los pronósticos, el cual se describe en la figura 10.

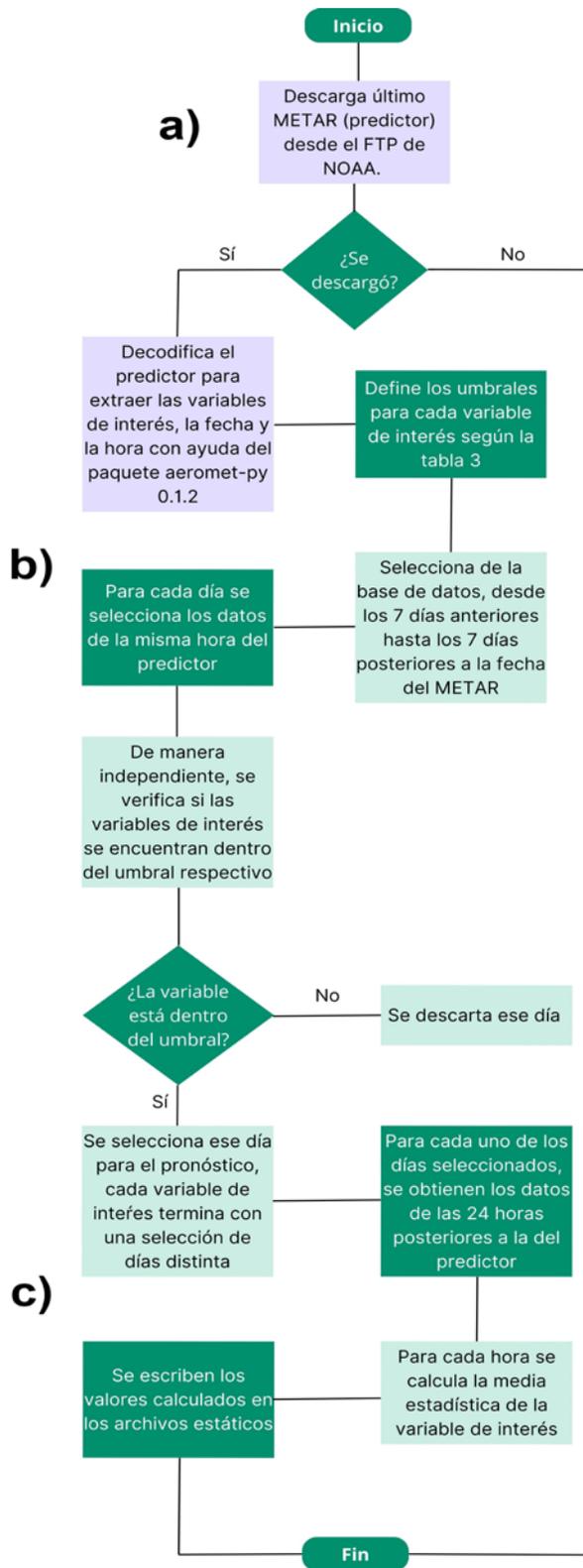


Figura 10. Diagrama de flujo que describe los pasos realizados por el algoritmo de metar-forecast 0.0.1. A).

Para cada una de las estaciones del cuadro 2, el algoritmo inicia con la descarga de la última observación de superficie o METAR, que actuará como el predictor (figura 12a). Si por alguna razón la observación no se puede descargar, como por ejemplo la falta de conexión a internet, el algoritmo finaliza sin realizar ningún cálculo. En caso contrario, se procede a decodificar el predictor (ver Figura 12b) para obtener las variables de interés, así como la fecha y la hora actuales. De este modo, en la base de datos se selecciona desde el séptimo día anterior, hasta el séptimo día posterior a la fecha actual. Seguidamente, para cada una de las variables de interés se procede a encontrar los días en los cuales el METAR de la misma hora del predictor indique valores dentro de los umbrales del cuadro 3, cada variable de manera independiente. Si el valor no se encuentra dentro de dichos umbrales, se descarta el día.

Al finalizar este proceso, se obtiene un listado de días en los que las variables de interés indican valores cercanos (cada variable cuenta con una selección de días independiente de las demás variables). Para cada uno de los días seleccionados, se obtienen los datos de las 24 horas posteriores a la hora del predictor (figura 12c), y se calcula la media horaria. Con los cálculos efectuados, se termina por escribirlos en los archivos estáticos que serán servidos en la aplicación web.

En algunos casos, el algoritmo no encuentra coincidencias para alguna de las variables, por lo que no se puede realizar ningún cálculo. En estos casos se retorna el texto “No data”, como se muestra en la figura 13.

3. RESULTADOS Y DISCUSIÓN

3.1. Crea Carpetas 2.5

Desde su implementación inicial en 2016 el programa sufrió varios cambios incorporando

las sugerencias y observaciones de los pronosticadores, especialmente por la ocurrencia de bugs o problemas inesperados en tiempo de ejecución. La versión que se muestra en este trabajo es la más estable luego de pasar por este proceso natural de prueba y error de todo software operativo. La efectividad del programa fue sondeada con los pronosticadores para la elaboración de este artículo. Aunque las experiencias varían entre cada pronosticador, en términos generales se pudo determinar que el tiempo de elaboración de las carpetas pasó a ser de unos 15 a 30 minutos por turno. Comparado con el rango anterior (desde 1 hora hasta 2 horas y media en el caso más extremo) esto significó una reducción máxima aproximada del 84% en cuanto al tiempo empleado para esta tarea.

Con esto el programa tiene más de 7 años de aplicarse exitosamente en el AIJS con algunas interrupciones cortas temporales (unas cuantas horas) en algunos días debido a problemas de mantenimiento, por ejemplo, cuando terceros modifican los links de productos utilizados o cuando una mala manipulación del personal borra accidentalmente archivos necesarios para el funcionamiento del programa. Debido a estas experiencias, es importante notar que el mantenimiento regular del programa es necesario por lo que se requiere personal capacitado en la programación científica.

3.2. get-metars 0.5.2

Si se usa como ejemplo el Aeropuerto Internacional John F. Kennedy cuyo código OACI es KJFK (Organización de Aviación Civil Internacional, 2020), la aplicación realiza una petición al sitio

web de Ogimet.com, la cual responde con los datos solicitados en forma de un documento HTML, get-metars recibe el documento y lleva a cabo un escaneo de los datos de interés con el fin de extraerlos y almacenarlos en archivos de texto dentro del disco duro local de la PC, a esto se le llama web scraping (Lawson, 2015).

Es importante destacar qué, el sitio web Ogimet.com solamente admite peticiones de datos de 31 días como máximo, por lo cual get-metar cuenta con un sistema de advertencia, en el caso que el usuario defina un intervalo de fechas que dé como resultado una cantidad de días mayor a 31 días. La figura 11 muestra el ejemplo de uso con KJFK, nótese que get-metars no tiene inconvenientes con los códigos OACI si se escriben en minúsculas, se define la fecha inicial de la petición desde el 01 de abril del año 2024 y se especifica con la opción -o que se quiere un METAR por línea dentro del archivo.

El archivo de texto con los METAR obtenidos mediante web scraping usando el comando mostrado en la figura 11 se presenta en la figura 12. Los archivos de texto son fácilmente manipulables mediante cualquier lenguaje de programación, por eso es que se elige este formato de salida para los datos.

3.3. metar-forecast 0.0.1

El resultado del pronóstico hecho por metar-forecast 0.0.1 se presenta en la figura 13. El propósito de este resultado es identificar un sesgo en los pronósticos al compararlos con las observaciones, y de este modo, realizar un mejor estimado de los datos que podrían registrarse en

```
> get-metars --init 2024-04-01 -o kjfk
Request from 2024-04-01 00:00:00 to 2024-05-01 23:59:00.
744 METAR requested succesfully.
```

Figura 11. Ejemplo de uso de get-metars 0.5.2 y su salida mostrada al usuario.

```

Archivo  Editar  Selección  Ver  Ir  Herramientas  Preferencias  Ayuda
Nuevo  Abrir  Guardar  Guardar como  Deshacer  Rehacer
1  202404010051 METAR KJFK 010051Z 28008KT 10SM BKN240 13/02 A2996 RMK AO2 SLP145 T01280022 $=
2  202404010151 METAR KJFK 010151Z 28007KT 10SM BKN230 13/01 A2997 RMK AO2 SLP147 T01330011 $=
3  202404010251 METAR KJFK 010251Z 26005KT 10SM BKN250 13/01 A2997 RMK AO2 SLP149 T01280006 51011 $=
4  202404010351 METAR KJFK 010351Z 25004KT 10SM SCT160 BKN200 10/04 A2999 RMK AO2 SLP155 T01000039 $=
5  202404010451 METAR KJFK 010451Z 00000KT 10SM SCT150 OVC200 10/04 A2999 RMK AO2 SLP154 T01000044 401670072 $=
6  202404010551 METAR KJFK 010551Z 00000KT 10SM BKN150 BKN220 09/04 A3000 RMK AO2 SLP159 T00890044 10144 20089 51010 $=
7  202404010651 METAR KJFK 010651Z 13003KT 10SM BKN140 OVC200 09/04 A2998 RMK AO2 SLP152 T00940044 $=
8  202404010751 METAR KJFK 010751Z 08004KT 10SM OVC140 09/04 A2997 RMK AO2 SLP148 T00940044 $=
9  202404010851 METAR KJFK 010851Z 08005KT 10SM SCT120 OVC160 09/03 A2997 RMK AO2 SLP149 T00940033 55010 $=
10 202404010951 METAR KJFK 010951Z 09005KT 10SM BKN110 OVC200 10/02 A3001 RMK AO2 SLP161 T01000017=
11 202404011051 METAR KJFK 011051Z 07005KT 10SM BKN100 BKN200 09/01 A2998 RMK AO2 SLP153 T00940011=
12 202404011151 METAR KJFK 011151Z 09006KT 10SM BKN100 OVC180 09/01 A3000 RMK AO2 SLP159 T00940011 10100 20083 51010=
13 202404011251 METAR KJFK 011251Z 09005KT 10SM FEW033 OVC100 10/02 A2999 RMK AO2 SLP154 T01000017=
14 202404011351 METAR KJFK 011351Z 08004KT 10SM -RA FEW030 OVC075 10/03 A2999 RMK AO2 RAB08 SLP153 P0000 T01000033=
15 202404011451 METAR KJFK 011451Z 06004KT 10SM -RA FEW037 OVC075 11/04 A2998 RMK AO2 SLP151 P0001 60001 T01060044 56008=
16 202404011551 METAR KJFK 011551Z 08003KT 10SM -RA FEW035 BKN075 OVC110 11/06 A2995 RMK AO2 SLP143 P0000 T01060056=
17 202404011651 METAR KJFK 011651Z 16005KT 10SM FEW042 BKN075 OVC100 11/06 A2993 RMK AO2 RAE29 SLP134 P0000 T01110061=
18 202404011751 METAR KJFK 011751Z VRB03KT 10SM FEW040 BKN055 OVC120 12/07 A2990 RMK AO2 RAB1652E03 SLP123 P0000 60001 T0
19 202404011851 METAR KJFK 011851Z 36004KT 10SM FEW032 BKN055 OVC120 11/05 A2990 RMK AO2 SLP123 T01110050=
20 202404011951 METAR KJFK 011951Z 17007KT 10SM FEW032 OVC055 11/06 A2991 RMK AO2 SLP127 T01060061=
21 202404012051 METAR KJFK 012051Z 18003KT 10SM BKN055 OVC075 10/06 A2993 RMK AO2 SLP133 T01000061 53010=
22 202404012151 METAR KJFK 012151Z 19003KT 10SM BKN055 OVC075 10/06 A2993 RMK AO2 SLP135 T01000061=
23 202404012251 METAR KJFK 012251Z 18005KT 10SM SCT055 BKN075 BKN250 09/07 A2994 RMK AO2 SLP137 T00940067=
24 202404012351 METAR KJFK 012351Z 22004KT 10SM BKN075 09/06 A2996 RMK AO2 SLP145 T00890061 10117 20089 53012=
25 202404020051 METAR KJFK 020051Z 00000KT 10SM OVC080 09/07 A2999 RMK AO2 SLP154 T00890067 $=
26 202404020151 METAR KJFK 020151Z 17004KT 10SM BKN080 09/07 A2999 RMK AO2 SLP155 T00890067 $=
27 202404020251 METAR KJFK 020251Z 19003KT 9SM FEW075 BKN250 08/06 A2999 RMK AO2 SLP153 T00780061 50008 $=
28 202404020351 METAR KJFK 020351Z 20005KT 6SM BR FEW070 SCT250 07/06 A2999 RMK AO2 SLP154 T00670056 $=
29 202404020451 METAR KJFK 020451Z 19004KT 9SM FEW075 BKN250 07/06 A2998 RMK AO2 SLP152 T00720061 401170067 $=
30 202404020551 METAR KJFK 020551Z 14003KT 8SM FEW080 BKN250 07/06 A2995 RMK AO2 SLP142 T00670056 10089 20067 58011 $=
31 202404020651 METAR KJFK 020651Z 17003KT 6SM BR BKN250 07/06 A2995 RMK AO2 SLP140 T00670061 $=
32 202404020751 METAR KJFK 020751Z 00000KT 7SM SCT180 BKN250 08/07 A2995 RMK AO2 SLP141 T00830072 $=
33 202404020851 METAR KJFK 020851Z 00000KT 6SM BR SCT110 BKN250 07/06 A2998 RMK AO2 SLP150 T00720061 53008 $=
34 202404020951 METAR KJFK 020951Z 08004KT 7SM -RA BKN070 OVC100 08/07 A2997 RMK AO2 RAB41 SLP147 P0000 T00780067 $=

```

Figura 12. Archivo de texto resultante de ejecutar get-metars, METAR's del mes de abril de 2024 para el Aeropuerto Internacional John F. Kennedy, USA.

Station: Aeropuerto Int. Juan Santamaría (MROC)

Forecast issued at: 2024/07/02 11:10Z

Observation (METAR) used as predictor: MROC 021100Z 00000KT 9999 FEW010 SCT180 20/18 A3001 NOSIG

Hour (UTC)	Wind Direction (°)	Wind Speed (kt)	Wind Gust (kt)	Temperature (°C)	Dewpoint (°C)	QNH (inHg)
12Z	70	5	No data	20.3	17.8	30.01
13Z	90	4	No data	21.9	18.3	30.03
14Z	120	4	No data	23.7	18.5	30.04
15Z	170	6	No data	24.9	18.6	30.04
16Z	180	6	No data	25.8	18.7	30.04
17Z	200	7	No data	26.2	19.0	30.04
18Z	200	8	No data	26.2	19.3	30.02
19Z	220	8	No data	25.9	19.7	30.00
20Z	220	8	No data	25.1	19.9	29.99
21Z	190	7	No data	24.0	19.9	29.98
22Z	200	6	No data	23.2	19.8	29.98
23Z	210	5	No data	22.4	19.9	29.99
00Z	170	4	No data	21.7	19.6	30.01
01Z	120	4	No data	21.4	19.6	30.02
02Z	130	4	No data	21.3	19.3	30.04
03Z	110	4	No data	21.2	19.2	30.04
04Z	90	4	No data	21.2	19.0	30.05
05Z	110	4	No data	20.9	18.8	30.05
06Z	100	4	No data	20.8	18.6	30.03
07Z	00	4	No data	20.6	18.3	30.01
08Z	100	5	No data	20.5	18.1	30.00
09Z	90	5	No data	20.3	17.9	30.00
10Z	80	5	No data	20.1	17.8	29.99
11Z	40	4	No data	20.0	17.7	30.00

Figura 13. Tabla que muestra los resultados del pronóstico para las próximas 24 horas. Se usó como predictor el METAR del Aeropuerto Internacional Juan Santamaría de las 05:00 pm (23:00UTC) del 09 de junio de 2023. «No data» indica que no se encontraron días que coincidan con el predictor.

Cuadro 4. Datos observados en el AIJS desde las 00:00 UTC hasta las 23:00 UTC del 10 de junio de 2024.

Variable	Error relativo promedio
Dirección del viento (°)	68°
Velocidad del viento (kt)	2.33 kt
Temperatura (°C)	0.71°C
Temperatura del punto de rocío (°C)	1.08°C
Presión QNH (inHg)	0.02 inHg

las próximas horas, de modo que puedan ser adjuntados en los productos generados por las OMA's, como por ejemplo las carpetas de vuelo aeronáuticas creadas con el programa Crea Carpetas 2.5.

En el cuadro 5 se muestran los datos observados en el AIJS, los cuales siguen la tendencia de los datos pronosticados de la figura 10. Se calculan los errores relativos horarios para cada una de las variables, excepto para las ráfagas de viento ya que no se obtuvieron valores del pronóstico y tampoco se observaron en la OMA, y se obtuvo un promedio de los errores absolutos como se muestra en el cuadro 4. Queda pendiente la recolección de datos pronosticados y observados para realizar un mejor análisis estadístico del modelo.

4. CONCLUSIONES

Se logró crear un programa especializado para la optimización de la elaboración de las carpetas aeronáuticas de despegue realizadas por la oficina de meteorología del aeropuerto internacional Juan Santamaría en San José, Costa Rica. Con el programa se logró disminuir el tiempo de realización de las carpetas en cada turno de pronóstico hasta un máximo del 84%. El programa se mantiene operativo desde el 2016 con algunos cambios ocasionales según sugerencias de los usuarios o corrección de errores en tiempo de ejecución. La versión más reciente se mantiene estable sin cambios desde el 2022.

La aplicación de línea de comandos get-metars 0.5.2 ha minimizado el tiempo de recolección de reportes de tipo METAR a menos de 6 minutos por cada año de datos, los cuales se han usado en otros productos generados en las OMA's. Aunque aún no se ha publicado una versión estable, se sigue probando, se realizan cambios y se agregan nuevas características conforme los productos generados en las Oficinas de Meteorología Aeronáutica lo requieran.

Desde que metar-forecast 0.0.1 se puso en operación, el personal de la Oficina de Meteorología Aeronáutica del AIJS cuenta con una herramienta más para la elaboración de las carpetas aeronáuticas confeccionadas para los vuelos internacionales. La información proporcionada por metar-forecast 0.0.1 guía al pronosticador hasta identificar un sesgo en el pronóstico, al compararlo con los datos observados. Con esta información el pronosticador puede identificar los posibles valores de las variables que se agregan a la información del vuelo, que se ingresan en el programa Crea Carpetas 2.5 con el cual se genera el producto final.

El código fuente de los programas se logró poner disponible para la descarga libre en la plataforma de gestión y organización de proyectos de programación de Github.

Estos programas especializados fueron desarrollados con relativa facilidad para cubrir necesidades operativas en materia de meteorología aeronáutica en los aeropuertos internacionales de Costa Rica. Las características que posee Python permitieron realizar esta labor por los meteorólogos del IMN sin necesidad de la intervención de un especialista en desarrollo de software, mediante un enfoque mayoritariamente autodidacta y empírico.

Entre estas características que favorecieron la escritura del código de los programas se destacan: la existencia de una amplia comunidad

Cuadro 5. Datos observados en el AIJS desde las 00:00 UTC hasta las 23:00 UTC del 10 de junio de 2024.

Hora (UTC)	Dirección del viento (°)	Velocidad del viento (kt)	Ráfagas de viento (kt)	Temperatura (°)	Punto de rocío (°)	QNH (inHg)
12Z	70.0	1	NaN	20	18	30.02
13Z	80.0	1	NaN	22	19	30.04
14Z	VARIABLE	2	NaN	25	19	30.03
15Z	230.0	7	NaN	24	21	30.04
16Z	240.0	7	NaN	25	21	30.04
17Z	260.0	7	NaN	27	21	30.04
18Z	240.0	9	NaN	26	22	30.02
19Z	250.0	11	NaN	26	22	30.00
20Z	280.0	9	NaN	24	22	29.99
21Z	250.0	8	NaN	21	21	29.97
22Z	160.0	5	NaN	21	21	29.98
23Z	220.0	6	NaN	21	21	30.02
00Z	250.0	5	NaN	21	20	30.05
01Z	220.0	3	NaN	21	20	30.07
02Z	300.0	2	NaN	21	20	30.09
03Z	340.0	3	NaN	21	20	30.09
04Z	0.0	0	NaN	21	19	30.09
05Z	0.0	0	NaN	21	20	30.09
06Z	0.0	0	NaN	20	20	30.06
07Z	0.0	0	NaN	20	20	30.02
08Z	0.0	0	NaN	20	20	30.00
09Z	080.0	3	NaN	20	19	29.98
10Z	0.0	0	NaN	19	18	29.98
11Z	0.0	0	NaN	19	18	30.00

en distintas plataformas en línea para la consulta (ejemplo, stackoverflow.com), así como amplia documentación oficial de fácil acceso para Python y sus distintas librerías. Esto permitió resolver todos los problemas de implementación de código que se generaron durante el desarrollo. Además, se destaca la existencia de herramientas para la instalación sencilla de las librerías para la elaboración de los programas (ejemplo: Anaconda), la existencia de potentes herramientas para el diseño de interfaces gráficas (ejemplo:

QT Designer) y para la escritura y depuración de código (ejemplo: Spyder).

Sin embargo, este tipo de enfoque autodidacta también trae desventajas entre las que se pueden mencionar: el mayor tiempo de desarrollo del software, posibles malas prácticas de programación que dificultan la lectura o modificación del código por terceros y la utilización de librerías o técnicas que pueden ser obsoletas (o estar próximas a serlo) lo que dificulta la incorporación de nuevas

características al software conforme aparecen nuevas necesidades o herramientas más potentes.

Esta es una experiencia que, con la capacitación correcta, puede ser replicada en otros servicios meteorológicos del mundo, y lo cual podría ser de mucha utilidad en países en vías de desarrollo o con condiciones similares a las de Costa Rica.

En este sentido lo autores sugieren incluir la programación científica en los talleres de capacitación de organismos internacionales como la OMM, USAID o Cooperación Española entre otros. Organismos que suelen llevar a cabo capacitaciones en materia de meteorología para países en vías de desarrollo.

5. REFERENCIAS

- Anaconda Software Distribution. (2020). Anaconda Documentation. *Anaconda Inc.* Recuperado de <https://docs.anaconda.com/>
- Ayer, V. M., Miguez, S., & Toby, B. H. (2014). Why scientists should learn to program in Python. *Powder Diffraction*, 29(S2), S48–S64. doi:10.1017/S0885715614000931
- Bennett, A. R., Hamman, J. J., & Nijssen, B. (2020). MetSim: A Python package for estimation and disaggregation of meteorological data. *c*, 5(47), 2042.
- Bezek, S., Coppa, M. J., & Di Bernardi, C. A. (2023). Análisis de accidentes aéreos con factor contribuyente meteorológico-región SAM OACI-periodo 2013-2022. In *VII Jornadas de Investigación, Transferencia y Extensión de la Facultad de Ingeniería* (La Plata, 25 al 27 de abril de 2023).
- Chacon, S., & Straub, B. (2014). *Pro git*. Apress.
- España Alvarado, S. (2022). Estudio del comportamiento de las empresas del sector aeronáutico en los mercados bursátiles en pandemia y postpandemia. Universidad Cooperativa de Colombia, Facultad de Ciencias Económicas, Administrativas y Contables, Comercio Internacional, Bogotá. Disponible en: <https://hdl.handle.net/20.500.12494/45913>
- Lawson, R. (2015). *Web scraping with Python*. Packt Publishing Ltd.
- Malvé, P. A., Nadal Mora, V. J., Pezzotti, S., & Di Bernardi, C. A. (2021). La automatización de la meteorología aeronáutica en la gestión del tránsito aéreo y aeroportuario. In *VI Jornadas de Investigación, Transferencia, Extensión y Enseñanza (ITEE)* (La Plata, 14 al 16 de septiembre de 2021).
- May, R. M., Goebbert, K. H., Thielen, J. E., Leeman, J. R., Camron, M. D., Bruick, Z., ... & Marsh, P. T. (2022). MetPy: A meteorological Python library for data analysis and visualization. *Bulletin of the American Meteorological Society*, 103(10), E2273-E2284.
- Méndez Frades, A., & Alcázar Izquierdo, Á. (2018). Meteorología aeronáutica: un marco de estudio esencial para comprender el impacto del tiempo en la aviación. *XXXV Jornadas Científicas de la Asociación Meteorológica Española 2018*. Disponible en: <http://hdl.handle.net/20.500.11765/11491>
- Mitlöhner, J., Neumaier, S., Umbrich, J., & Polleres, A. (2016, August). Characteristics of open data CSV files. In *2016 2nd International Conference on Open and Big Data (OBD)* (pp. 72-79). IEEE.
- Moreno García, M. D. C., & Gil Aguinaliu, M. Á. (2003). Análisis de la siniestralidad aérea por causa meteorológica (1970-1999). *Investigaciones geográficas*, nº 30, 2003; pp. 7-25.
- Organización de Aviación Civil Internacional (2020). Manual on Location Indicators (Doc 7910/178). Recuperado de: <https://store.icao.int/en/location-indicators-doc-7910>
- Organización de Aviación Civil Internacional (2018). Anexo 3: Servicio meteorológico para la navegación aérea internacional. Recuperado de: <https://elibrary.icao.int/home>

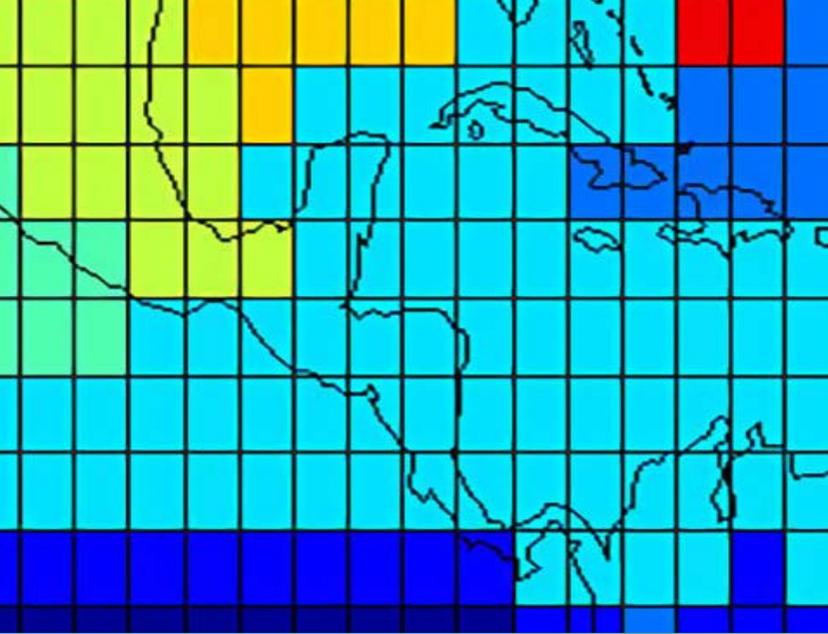
Organización Meteorológica Mundial (2003). *Guía de prácticas para oficinas meteorológicas al servicio de la aviación, OMM-N°732*. Recuperado de: <https://library.wmo.int/idurl/4/29123>

Organización Meteorológica Mundial. (2018). *Guía de Instrumentos y Métodos de Observación (OMM-N° 8)*. Recuperado de: <https://library.wmo.int/idurl/4/68714>

Raybaut, P. (2009). *Spyder-documentation*. Disponible en: pythonhosted.org, 769.

Stack Overflow (2024) Stack Overflow Developer Survey 2024. Disponible en: <https://stackoverflow.blog/2024/07/24/developers-want-more-more-more-the-2024-results-from-stack-overflow-s-annual-developer-survey> (consultado el 10 de setiembre de 2024)

Valverde, J. P. (2017). Importancia de la interpretación e implementación de las regulaciones y procedimientos meteorológicos para la aviación internacional. *Tópicos Meteorológicos y Oceanográficos*. 16 (2) 19-27



Instituto Meteorológico Nacional

Sitio web:
www.imn.ac.cr

Teléfono: (506) 2222 5616

Apartado postal: 5583-1000
San José
Costa Rica

