

Journal of Computational Astronomy & Astronomical Computing

Revista de astronomía computacional y cálculo astronómico

Nº 2. Marzo 2025

In this issue:

Editorial

- 5** *El planisferio FAAE* Sergio Díaz
- 21** *Asteroide Apophis, “el destructor”* Juan Carlos Terradillos
- 61** *Caracterización de cúmulos estelares abiertos* Joaquín Álvaro
- 75** *Diseño, construcción y programación de dispositivos astronómicos: una introducción* Sergio Alonso y Javier Flores
- 87** *Astronomical Computing: Cálculo de ángulos de orientación de la Tierra y transformaciones de coordenadas* Tomás Alonso
- 105** *Software for Photometry & Astrometry: Astrometría con Tycho*
Ramón Naves y Montse Campàs

Edited by **Federación de Asociaciones Astronómicas de España** in Madrid (c/Serrano, 117)

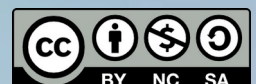
The Journal of Computational Astronomy & Astronomical Computing is an effort by the FAAE - Grupo de Cálculo Astronómico (GCA) to encourage the use of software tools and the development of codes and algorithms for astronomical applications within the framework of amateur astronomy, as well as to connect the amateur community with the professional astronomy community and promote cross-collaboration and ProAm projects between both groups.



Sociedad Española de
Astronomía



Federación de Asociaciones
Astronómicas de España



This work is licensed under a Creative
Commons Attribution 4.0 License



Journal of Computational Astronomy and Astronomical Computing

An online publication of Federación de Asociaciones Astronómicas de España, Grupo de Cálculo Astronómico

Editorial Board: Enrique Velasco, Sergio Díaz, Joaquín Álvaro, César González, José Luis Navarro, Enrique Sánchez, Roberto Ruiz, Ramón Naves y Montse Campàs.

Advisory Board: Luis Mederos and Matilde Fernández.

<https://federacionastronomica.es/index.php/recursos-astronomicos/journal-of-ca-ac>

EDITORIAL

Con ocasión del lanzamiento de este segundo número de la revista JCAAC, quisiera en primer lugar dar las gracias por la buena acogida que ha tenido el primer número de la revista en la comunidad astronómica, con más de mil bajadas completas. Reiteramos nuestro deseo de que JCAAC se convierta en un foro que sirva para diseminar trabajos realizados por astrónomos aficionados y profesionales, y animamos a cualquier miembro de la comunidad a enviar sus contribuciones, que sin duda serán valiosas aportaciones al desarrollo de la computación en el marco de la astronomía.

En este número presentamos un buen número de nuevos artículos, todos ellos muy interesantes y de gran calidad. Iniciamos dos nueva secciones, dedicadas al uso de herramientas del Observatorio Virtual, con interesantes aplicaciones en astrofísica, y al desarrollo de software específico para el control de dispositivos en astronomía. Además, contamos con el primer artículo de una serie de ellos que estarán dedicados al asteroide Apophis, considerado peligroso por la posibilidad de que pueda colisionar con la Tierra en un futuro, y con otro dedicado al Planisferio FAAE, una herramienta que permite la construcción manual de un planisferio muy completo para su uso en el campo o como accesorio práctico en la preparación de observaciones. Finalmente, contamos con nuevas e interesantes entregas en las secciones de cálculo astronómico y de software para fotometría y astrometría; en esta última, se explica el uso del software Tycho, una potente herramienta astrométrica para el astrónomo amateur y profesional.

On the occasion of the launch of this second issue of the JCAAC journal, I would first like to express my gratitude for the warm reception of the first issue within the astronomical community, with more than a thousand complete downloads. We reaffirm our hope that JCAAC will become a forum for disseminating work carried out by both amateur and professional astronomers, and we encourage all members of the community to submit their contributions, which will undoubtedly be valuable additions to the development of computational methods in the field of astronomy.

In this issue, we present a wide selection of new articles, all of great interest and high quality. We are introducing two new sections: one dedicated to the use of Virtual Observatory tools, with fascinating applications in astrophysics, and another focused on the development of specialised software for controlling astronomical devices. In addition, we feature the first in a series of articles dedicated to the asteroid Apophis, considered hazardous due to its potential future collision with Earth, as well as another article on the FAAE Star Finder, a tool that enables the manual construction of a highly comprehensive planisphere for field use or as a practical accessory for observation planning. Finally, this issue includes new contributions in the sections on astronomical calculations and software for photometry and astrometry. In the latter, we present an explanation of how to use Tycho, a powerful astrometric tool for both amateur and professional astronomers.

E. Velasco, February 2025

CONTENTS

Editorial

Applications

<i>El planisferio FAAE</i> , Sergio Díaz	5
<i>Asteroide Apophis, "el destructor"</i> , Juan Carlos Terradillos	21
Section: Herramientas del observatorio virtual , Joaquín Álvaro Contreras	
<i>Caracterización de cúmulos estelares abiertos</i>	61
Section: Programación de dispositivos astronómicos , Sergio Alonso y Javier Flores	
<i>Diseño, construcción y programación de dispositivos astronómicos: una introducción</i>	75
Section: Astronomical Computing , Tomás Alonso	
<i>Cálculo de ángulos de orientación de la Tierra y transformaciones de coordenadas</i>	87
Section: Software for Photometry & Astrometry , Ramón Naves y Montse Campàs,	
<i>Astrometría con Tycho</i>	105

All contents of this publication are licensed under a Creative Commons Attribution 4.0 License
Published codes can be used and distributed under a Gnu gpl license



With the sponsorship of Sociedad Española de Astronomía (SEA), Grupo ProAm

APPLICATIONS

El planisferio FAAE

Sergio Díaz Ruiz¹¹Asociación Astronomía Sevilla, Spain. E-mail: sergio.diaz.ruiz@gmail.com.**Keywords:** planisferio celeste, catálogo, proyección geográfica, python, cartopy

© Este artículo está protegido bajo una licencia Creative Commons Attribution 4.0 License

Este artículo adjunta un software accesible en https://github.com/JCAAC-FAAE/No02-Mar2025-EL_planisferio_FAAE

Resumen

El planisferio FAAE es una aplicación web que permite generar planisferios personalizados para cualquier ubicación geográfica con latitud comprendida entre 20° a 66° norte o sur, y para la zona horaria indicada. Al tener en cuenta la longitud geográfica y la zona horaria, la aplicación corrige automáticamente el desfase horario en los planisferios que genera, por lo que se pueden usar con la hora local, a diferencia de los planisferios clásicos, que están rotulados en Tiempo Universal. En el presente artículo se describe la arquitectura de la aplicación así como los aspectos clave para su desarrollo, incluyendo las fuentes de datos utilizadas, los procedimientos de cálculo aplicados y los detalles de implementación más relevantes. A este respecto, el artículo va acompañado de un *notebook Jupyter* [1] que permite reproducir los elementos más significativos del planisferio con una extensión mínima y reduciendo en lo posible las dependencias a librerías externas.

Abstract

The FAAE planisphere is a web application that generates customized planispheres for any geographic location with latitude between 20° to 66° north or south, and for the time zone indicated. By taking into account the geographical longitude and the time zone, the application automatically corrects the time offset in the planispheres it generates, so that they can be used with local time, unlike the classic planispheres, which are labeled in Universal Time. This article describes the architecture of the application as well as the key aspects for its development, including the data sources used, the calculation procedures applied and the most relevant implementation details. In this regard, the article is accompanied by a *Jupyter notebook* [1] that reproduces the most significant elements of the planisphere with a minimum extension and reducing as much as possible the dependencies to external libraries.

1. Introducción

Un planisferio celeste es una potente herramienta de cálculo astronómico que nos permite determinar el cielo visible a una fecha y hora dadas, para una ubicación geográfica determinada. Hoy en día, con la disponibilidad de aplicaciones de planetario para ordenador y en particular, para los siempre presentes móviles, los planisferios han pasado a un cierto segundo plano. Aunque la precisión que nos proporcionan las aplicaciones informáticas es mayor, el planisferio sigue presentando ventajas, entre las que podemos destacar:

- Facilita el aprendizaje de las constelaciones
- Permite entender de manera intuitiva el movimiento diurno
- Su “interfaz de usuario” es el más sencillo posible, teniendo sólo un control, el giro de la ventana de horizonte
- No requiere batería ni tiene mantenimiento

- Respetar la adaptación visual a la oscuridad, combinado con una linterna de luz roja regulable en intensidad

Por otro lado, su uso presenta ciertos inconvenientes:

- La distorsión debida a la proyección usada para representar las estrellas en el plano puede resultar al principio difícil de interpretar y de hacer corresponder con el cielo a simple vista
- Debe ser lo suficientemente grande (en torno a 30 cm de diámetro) para resultar cómodo al uso, de lo contrario la ventana de horizonte puede ser demasiado pequeña para apreciar correctamente el cielo visible
- Dado que los planisferios están calculados para el meridiano de Greenwich, es necesario compensar mentalmente la hora del reloj considerando tanto la zona horaria oficial en uso como la longitud geográfica

En particular, este último punto suele resultar especialmente incómodo para los nuevos usuarios del planisferio, o si se usa muy esporádicamente. En parte para mitigar este aspecto, en el seno del Grupo de Cálculo Astronómico de la Federación de Asociaciones Astronómicas de España (FAAE) se ha desarrollado el planisferio FAAE.

2. El planisferio FAAE

El planisferio FAAE es una aplicación web [2] que permite generar planisferios personalizados para cualquier ubicación geográfica con latitud comprendida entre 20° a 66° norte o sur¹, y para la zona horaria indicada.

Al tener en cuenta la longitud geográfica y zona horaria, la aplicación corrige automáticamente el desfase horario en los planisferios que genera, por lo que se pueden usar con la hora local (a diferencia de los planisferios clásicos, que están rotulados en Tiempo Universal).

Otras características importantes de los planisferios generados por la aplicación son los siguientes:

- el círculo horario coincide con la ascensión recta, al igual que el planisferio Procivel, lo que permite usar el disco fijo como carta estelar
- la eclíptica dispone de marcas diarias, lo que permite estimar la posición del Sol, y con ello, determinar aproximadamente los instantes de orto, culminación y ocaso solar
- el disco giratorio incorpora ventanas con límites en -6° , -12° y -18° de altura, lo que permite estimar el inicio y fin de los crepúsculos

Hay que tener en cuenta que en aquellos lugares donde existe horario de verano, como España, o bien manejamos dos planisferios, uno para horario de invierno y otro para verano, o bien tendremos que compensar esa diferencia de una hora. Por ejemplo, si usamos el planisferio "de invierno" durante el horario de verano, tendremos que restar mentalmente una hora a nuestro reloj.

La aplicación web del Planisferio FAAE está accesible en https://sergiodiaz.eu/planisferio_faae [2] y consiste esencialmente en un mapa que podemos arrastrar y ampliar y que nos permite seleccionar una localización. Esto despliega una ventana emergente que indica la latitud y longitud del lugar y permite seleccionar la zona horaria en caso de que dicha localización distinga entre horarios de invierno y verano, como es el caso de España. Por defecto estará activa la opción "Compensar desfase horario", generando un planisferio con la corrección horaria ya incorporada, aunque es posible desmarcarla para obtener un planisferio convencional, calculado para el meridiano de referencia (Greenwich, longitud geográfica 0°).

¹Para latitudes próximas al ecuador, el grado de distorsión de la proyección usada es demasiado alto y se recomienda el diseño de Toshimi Taki <https://www.educa-ciencia.com/planisferio-doble-cara.htm>, en el que el mapa estelar tiene dos caras, una por hemisferio celeste, y se consulta alternando una u otra en función de la dirección en la que observemos.

Por último, es posible seleccionar el idioma en que se representarán los meses, los puntos cardinales y el rótulo del planisferio.

Al pulsar sobre el botón “Generar planisferio” se descargará un fichero comprimido en formato ZIP que contiene tres documentos PDF, a imprimir a máxima calidad:

- `fiijo_(opaco).pdf`: disco “fijo” o mapa estelar, que imprimiremos en papel blanco.
- `giratorio_(transparente).pdf`: disco “giratorio” o ventana de horizonte, que idealmente imprimiremos en soporte transparente, p.ej. láser sobre acetato.
- `Planisferio_FAAE_Instrucciones.pdf`: disco para colocar en el reverso del disco fijo, con un resumen de las instrucciones de uso.

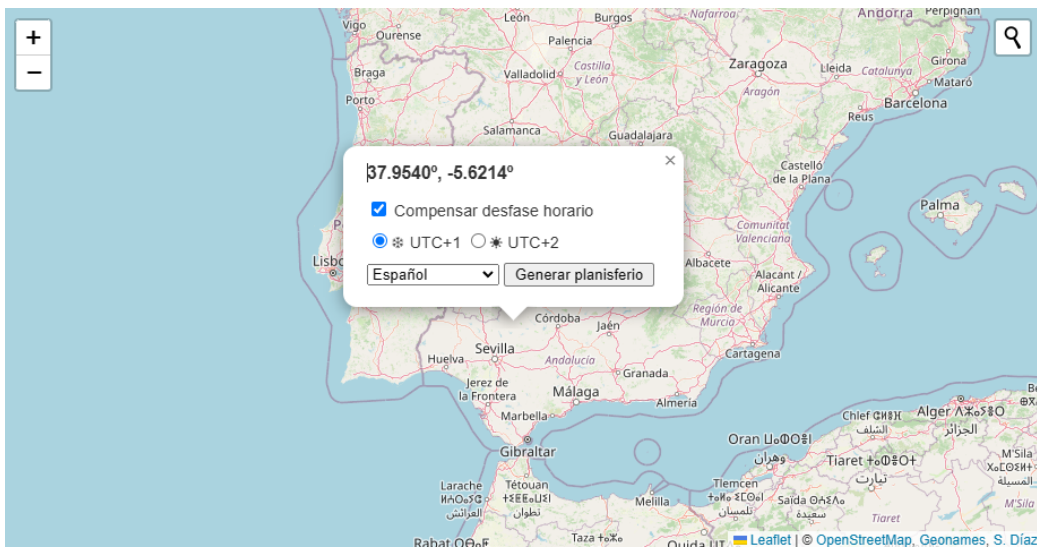


Figura 1. Aplicación web Planisferio FAEE.

Para la realización de talleres formativos en torno al planisferio FAEE se ha elaborado una guía didáctica [3] con ejercicios y material complementario.

3. El disco fijo

El disco fijo, Fig. 2, está formado esencialmente por un mapa estelar, centrado en el polo celeste visible desde la ubicación seleccionada. Esto se aprecia fácilmente al estar representada la rejilla de coordenadas ecuatoriales, que en la proyección usada se compone de líneas radiales de ascensión recta constante y circunferencias concéntricas con declinación constante, quedando destactada la correspondiente al ecuador celeste. El rango de declinaciones del mapa coincide con el rango visible desde la ubicación. La leyenda correspondiente a la ascensión recta se expresa en horas y se coloca en disposición de anillo alrededor del mapa, formando el círculo horario, que nos permitirá, junto con el disco giratorio, seleccionar la fecha y hora de observación. Este doble rol de leyenda de ascensiones rectas y círculo horario está heredado del diseño del planisferio Procivell [4].

El interior del disco representa las estrellas brillantes, hasta la magnitud 4.5, que son visibles desde la ubicación geográfica seleccionada. Para facilitar el reconocimiento del cielo visible, se han trazado los asterismos y etiquetado las constelaciones con la abreviatura latina oficial. Las estrellas más destacadas

están etiquetadas usando su denominación de Bayer.

Se incluye una representación de la Vía Láctea con 5 niveles de brillo (isofotas), para distinguir las zonas más brillantes, y los objetos de cielo profundo de mayor extensión apreciables a simple vista bajo cielos suficientemente oscuros.

El mapa también incorpora la proyección de la eclíptica sobre el fondo de estrellas, quedando representada como una curva cerrada con marcas diarias que permiten estimar la posición del Sol cualquier día del año, y con ello, el orto, culminación y ocaso solar como si se tratase de cualquier otra estrella del mapa. Para evitar que el etiquetado de dichas marcas congestione esta zona del mapa estelar, se ha recurrido a añadir marcas de distintos grosores y longitud los días 1, 15 y múltiplos de 5 de cada mes, etiquetando únicamente el día 1 de los meses de marzo, junio, septiembre y diciembre, con numeros romanos.

La línea de eclíptica también indica el camino aproximado que siguen los planetas en el cielo, si bien el planisferio no incluye ningún mecanismo para estimar su posición.

4. El disco giratorio

El siguiente elemento del planisferio, que irá colocado sobre el disco fijo, contiene una ventana que nos mostrará el cielo visible en una fecha y hora dadas, Fig. 3. Esta ventana está delimitada por la línea del horizonte, que en esta proyección adopta una forma achatada para las latitudes medias, siendo perfectamente circular únicamente en los polos geográficos.

La ventana contiene la rejilla de coordenadas horizontales, formada por los almicanarats y los verticales cada 15° , además de los puntos cardinales. Se han resaltado las líneas de horizonte, meridiano local y el primer vertical. En la zona del cenit se ha evitado la concentración de verticales, manteniendo únicamente el cruce entre el meridiano y el primer vertical. Las dos zonas contiguas a la ventana situadas en torno a los puntos cardinales este y oeste, corresponden a las franjas de los crepúsculos, delimitadas por el rango de acimuts dentro del cual se mueve el Sol. Se han marcado con líneas discontinuas los almicanarats de -6° , -12° y -18° , correspondientes a los límites de los crepúsculos civil, náutico y astronómico, respectivamente.

El círculo exterior o círculo de fechas está graduado con marcas diarias y segmentado por los meses del año. Está dimensionado de forma que su circunferencia exterior concuerda con la circunferencia interior del círculo horario del disco fijo.

El centro del disco dispone de una pequeña marca que servirá como eje central en el montaje.

Por último, además de mostrar las coordenadas geográficas y zona horaria para la que se ha calculado el planisferio, se incluye una leyenda con la escala de magnitudes de las estrellas tal como se han representado en el disco fijo.

La Fig. 4 muestra la variación de la forma de las ventanas de horizonte y crepúsculos para distintas latitudes. Aunque se muestran latitudes del hemisferio norte, el efecto es simétrico para el hemisferio sur. Se observa cómo la distorsión de la ventana es mayor conforme la latitud es más cercana al ecuador.

5. Arquitectura de la aplicación

El planisferio FAAE es una aplicación web compuesta por una capa cliente desarrollada en base a la librería de visualización de mapas `leaflet` [5] y una capa servidor responsable de la generación del planisferio. La arquitectura se ilustra en la Fig. 5.

La capa cliente está incrustada en una página web como un elemento `script` desarrollado en Javascript, en la que muestra el mapa que permite seleccionar la ubicación. La librería `leaflet` se encarga de los detalles de visualización e interacción con el mapa, incluyendo la descarga de las teselas necesarias desde el servidor de `OpenStreetMap`. El desarrollo realizado en la capa cliente hace uso del servicio

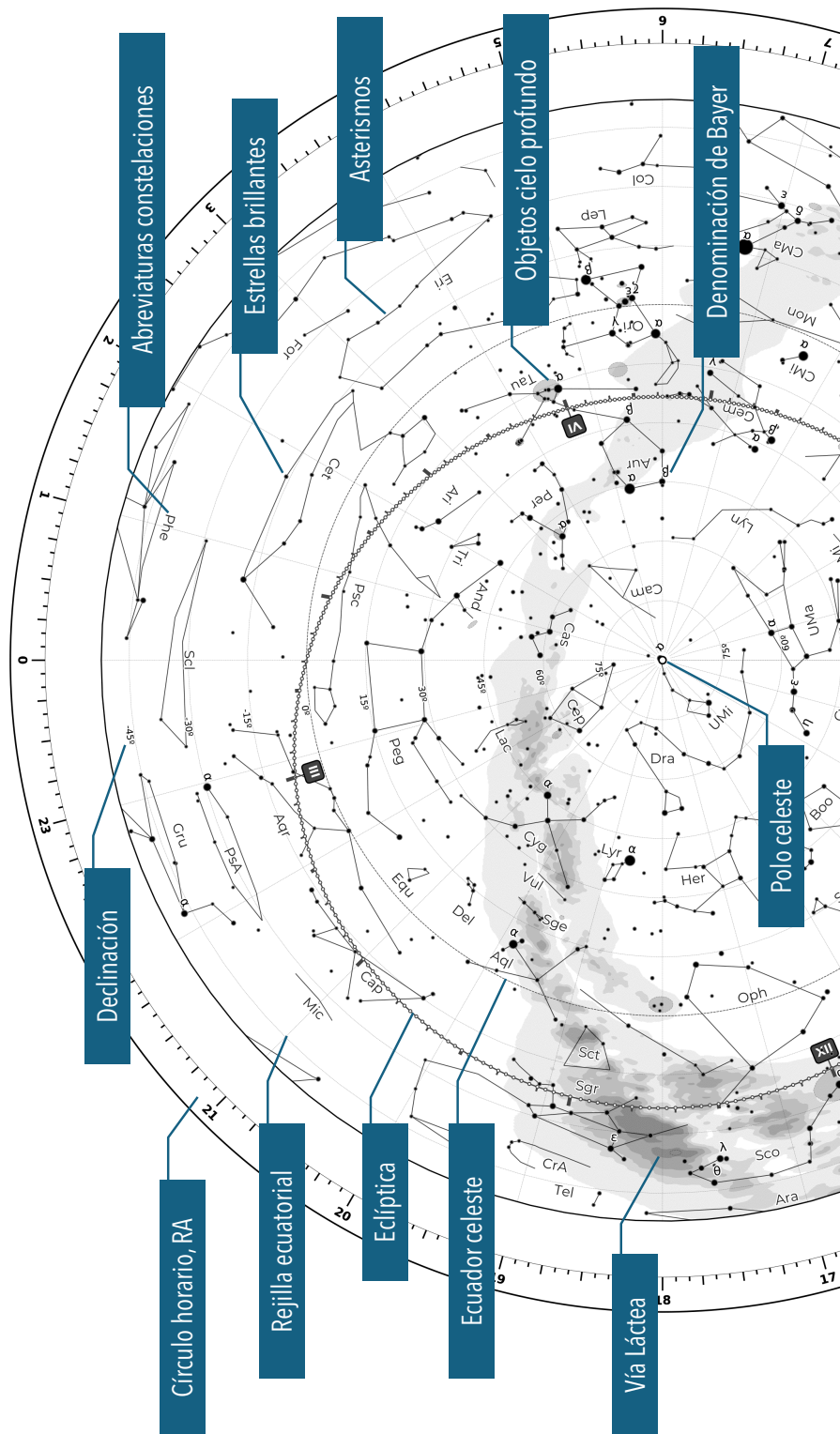


Figura 2. Disco fijo: mapa estelar.

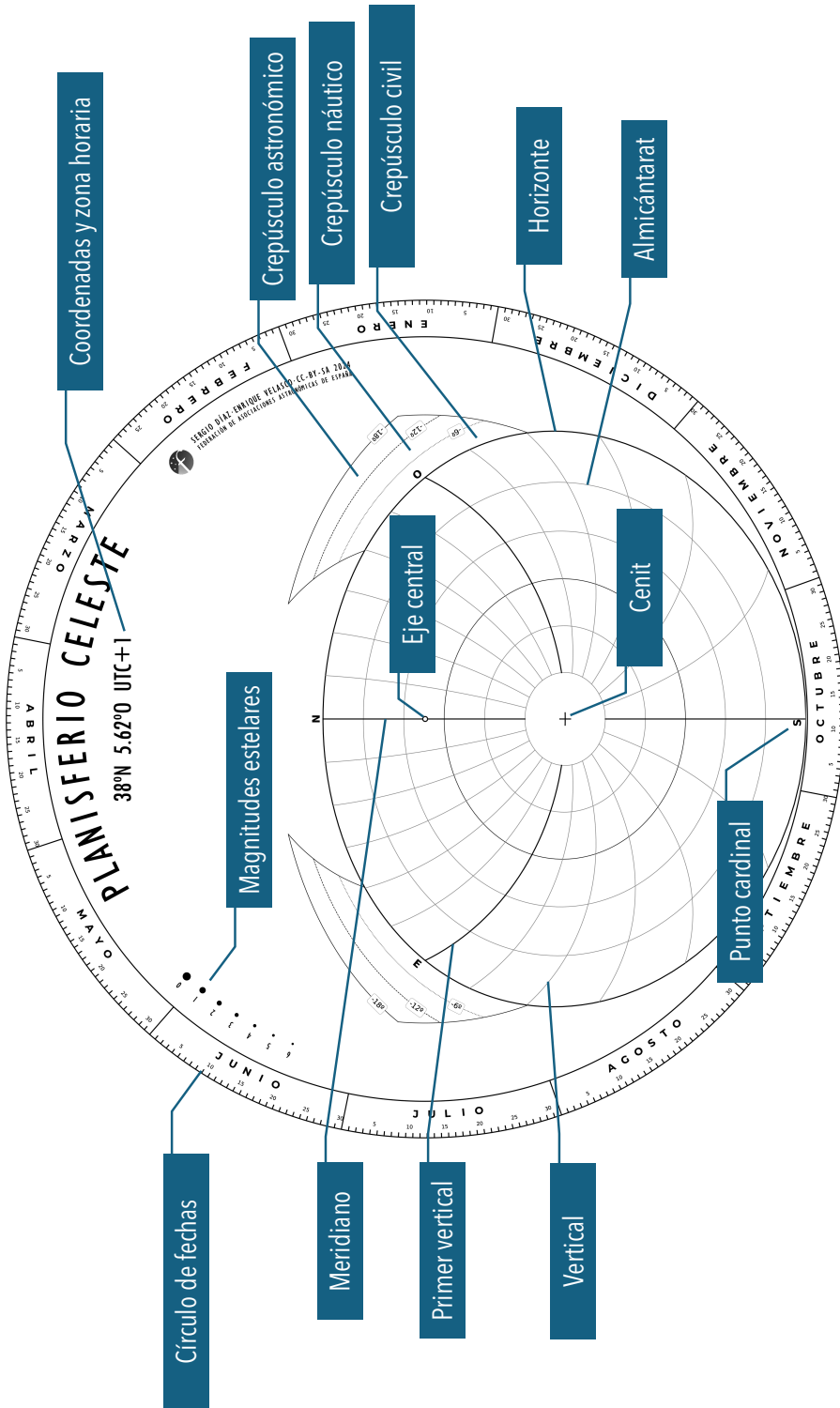


Figura 3. Disco giratorio: ventana del horizonte y crepúsculos.

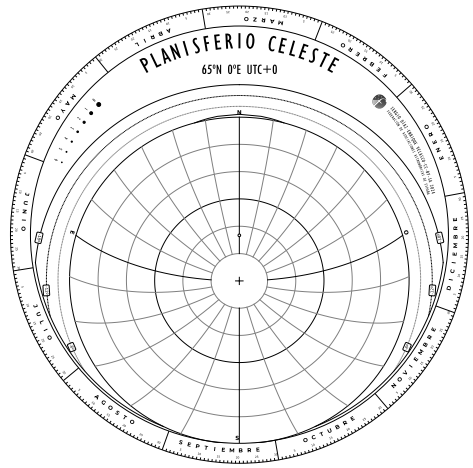
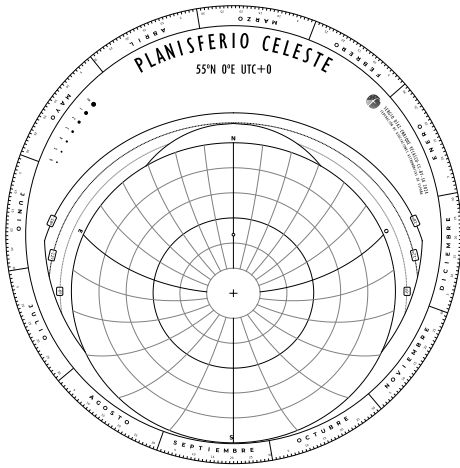
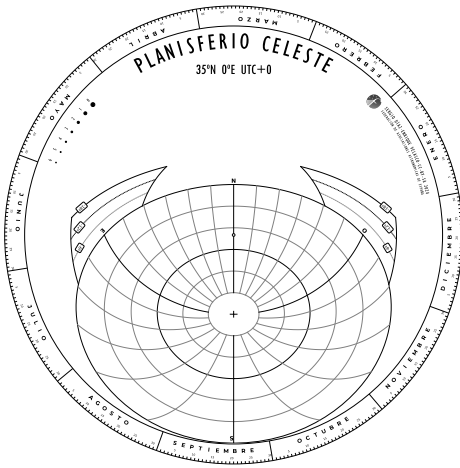


Figura 4. Discos giratorios calculados para distintas latitudes.

timezoneJSON de *geonames.org* para obtener, en formato JSON, la información sobre zona horaria aplicable al punto seleccionado por el usuario sobre el mapa.

La capa servidor está desarrollada en Python y presenta un microservicio con un único *endpoint*, usando la librería *flask*. Este *endpoint* hace uso de una librería de cálculo astronómico, de desarrollo propio, que en este caso se emplea para generar los discos fijo y giratorio en base a la selección del usuario. El *endpoint* se encarga de empaquetar ambos discos, exportados a formato PDF, junto con un tercer documento PDF con las instrucciones de uso, en un fichero comprimido ZIP que se devuelve al navegador como respuesta a su solicitud al microservicio.

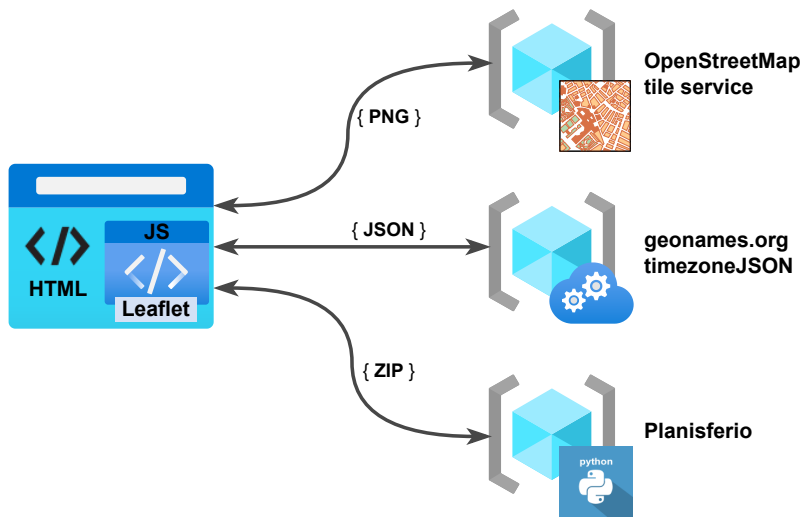


Figura 5. Arquitectura de la aplicación.

6. Fuentes de datos

El catálogo estelar básico para un planisferio es el *Yale Bright Star Catalog* [6], que incluye todas las estrellas hasta aproximadamente la magnitud 7. No obstante, la librería, de desarrollo propio, usada para generar los planisferios FAAE incorpora también funcionalidad para generar mapas celestes arbitrarios por lo que se ha utilizado un catálogo más completo, filtrándolo para el caso del planisferio. En este caso, se trata del catálogo ATHYG [7], que fusiona los catálogos *Tycho*, *Hipparcos*, *Yale Bright Star* y *Gliese*, mejorando los datos astrométricos con *Gaia DR3*.

El catálogo objetos de cielo profundo empleado es el suplementario del catálogo HYG [8], compilado por el mismo autor que el anterior. Se trata de una fusión de los catálogos NGC/IC, PGC y Collinder, entre otros, incorporando unos 220000 objetos.

Para los asterismos de las constelaciones se ha utilizado el conjunto de datos publicado por *Stellarium* como *cultura celeste moderna* [9], que incluye los segmentos que componen el asterismo de cada constelación.

Las geometrías de la Vía Láctea están basadas en el conjunto de datos incluido en la librería *D3.js*, en formato GeoJSON, que presenta las isofotas a cinco niveles de brillos superficial [10].

En el *notebook Jupyter* [1] que acompaña al artículo se muestra con detalle cómo hacer uso de estos

catálogos.

La selección de fuentes de datos se basa en la usada en el proyecto *Western Constellations Atlas of Space* de E. Lutz [11].

7. Implementación

El código fuente de la aplicación resulta extenso ya que gran parte se dedica a aspectos puramente gráficos y estéticos, que no se consideran de interés en este artículo. Por otro lado, se usan rutinas de efemérides que también quedan fuera del ámbito del mismo.

Por ello, el resto del artículo pone el foco en los distintos métodos de cálculo usados, mientras que los aspectos más relevantes de la implementación se concentran en un *notebook Jupiter* disponible como material suplementario [1]. Este *notebook* cubre el acceso y procesamiento de los catálogos, la representación de las estrellas, asterismos, la eclíptica y la Vía Láctea, la proyección de la ventana de horizonte y crepúsculos, y añade algunas indicaciones sobre optimización que son de especial interés a la hora de publicar la aplicación como servicio web.

La implementación se ha realizado en Python, usando la librería *cartopy* [12] para la representación gráfica. Esta librería incorpora todo lo necesario para el cálculo de las proyecciones cartográficas. Está construida a su vez sobre la librería de visualización de propósito general *matplotlib*, que usamos directamente para insertar todos aquellos elementos del planisferio que no involucran proyecciones, como es el caso de los círculos horario y de fechas, rótulos, etc.

8. Cálculo de los elementos del planisferio

8.1. Proyección cartográfica

Si bien en astronomía de posición se considera que todos los objetos celestes están proyectados en una esfera que rodea a la Tierra, obviando sus distancias verdaderas, para construir el planisferio necesitamos dar un paso más allá y aplicar una nueva transformación geométrica que permita representar estos objetos en un plano. Existen multitud de estas transformaciones, conocidas como *proyecciones cartográficas*, que se diferencian en qué características geométricas (área, forma, distancia, etc.) son capaces de conservar, a costa de deformar otras.

En el caso de los planisferios celestes, la proyección elegida debe cumplir los siguientes requisitos: (a) generar una representación circular, para permitir el giro de la ventana de horizonte sobre el disco fijo en función de la fecha y la hora, como reflejo de movimientos de rotación diurno y anual de la Tierra; (b) la ventana de horizonte debe tener una forma fija independientemente de la zona del cielo; y (c) permitir la representación de más de la mitad de la esfera celeste, ya que el cielo observable desde un lugar con latitud φ cubre el rango de declinaciones desde $+90^\circ$ hasta $-90^\circ + \varphi$ en el caso del hemisferio norte, y desde $90^\circ + \varphi$ hasta -90° en el caso del hemisferio sur.

El primer requisito implica que el disco fijo presente una forma similar a la de un reloj, con líneas de ascensión recta distribuidas radial y uniformemente, y situando el polo celeste visible en el centro. El segundo, obliga a que las líneas de declinación constante sean circunferencias concéntricas, centradas en el polo visible. En la práctica estas condiciones restringen a que la proyección usada sea de tipo acimutal. El tercer requisito descarta proyecciones tales como la ortográfica y la gnomónica.

Las proyecciones acimutales proyectan la superficie esférica sobre un plano tangente a la misma en un punto dado. En este caso, ese punto será el polo celeste visible, sobre el que se produce el giro aparente en el cielo. Conforme nos alejamos del punto tangente, el grado de distorsión aumenta. La Fig. 6 muestra el mapa estelar y la ventana resultante para tres de las proyecciones acimutales que cumplen los requisitos del planisferio, como son la estereográfica, la acimutal equidistante y la acimutal de áreas equivalentes

de Lambert. Sin entrar en las características de estas proyecciones, se puede observar a partir de la figura que la proyección estereográfica mantiene mejor las formas para las constelaciones más alejadas del polo visible a costa de reducir drásticamente el tamaño de las más próximas a él. Esto afecta también sustancialmente a la forma de la ventana, que presenta un grado de asimetría considerable. La situación opuesta se produce en el caso de la proyección de Lambert. Si bien en esta proyección la ventana tiene mayor tamaño que en el resto, lo que incrementa la zona útil del planisferio a la hora de estimar el cielo visible, el cenit no queda centrado en la ventana, lo que puede entorpecer su comparación con el cielo visible a simple vista. La proyección acimutal equidistante se presenta como solución de compromiso, con la ventaja de que la escala de distancias en declinación se mantiene constante en todo el mapa.

La proyección acimutal equidistante se corresponde con la siguiente representación polar de las coordenadas ecuatoriales, ascensión recta α y declinación δ :

$$\rho = R \left(\frac{\pi}{2} - |\delta| \right), \quad \theta = \frac{\pi}{2} - \alpha$$

donde ρ es la distancia al centro (siendo R un factor de escala, tal que el ecuador celeste tendrá un radio de $R\frac{\pi}{2}$) y θ , el ángulo polar (medido desde el eje x del plano). Esta expresión ya considera que el observador se encuentra en el interior de la esfera.

Sin embargo, en la implementación que acompaña el artículo, la proyección es aplicada por la librería cartográfica usada, `cartopy`, por lo que no es necesario tratar directamente con estas expresiones.

8.2. *Eclíptica*

Para poder trazar la eclíptica sobre el mapa estelar es necesario determinar las coordenadas ecuatoriales de un conjunto de puntos situados en ella, para después unirlos por segmentos, construyendo una aproximación a una curva cerrada.

Podemos obtener este conjunto de puntos de dos maneras:

- Trabajando en el sistema eclíptico medio, hacer un barrido rotando alrededor del eje Z un vector cualquiera situado en el ecuador eclíptico, p.ej. $[1, 0, 0]$. A continuación basta con transformar al sistema ecuatorial medio el conjunto de vectores resultante.
- Usar cualquier método de cálculo de efemérides que estime la posición del Sol, con un muestreo de periodicidad fija. En el caso del planisferio FAAE, el muestreo es diario, a las 12:00 UT.

El primer método es muy simple y eficiente, sin embargo, el segundo es más interesante ya que nos permite posicionar el Sol en la eclíptica cualquier día del año. Dada la precisión del planisferio, no es necesario recurrir a un cálculo riguroso de efemérides, y en cualquier caso, es posible precalcular todas las posiciones, evitando ralentizar el proceso de generación del planisferio.

Éste último es el procedimiento implementado en el *notebook* [1], si bien el cálculo de efemérides queda fuera del alcance del artículo y se ha sustituido por una tabla de valores.

8.3. *Proyección del sistema horizontal*

Las ventanas de horizonte y crepúsculos del disco giratorio, así como sus líneas auxiliares, los verticales y almicantrats, una vez proyectadas, son curvas trascendentes. Por ello, resulta más sencillo calcularlas en base a segmentos relativamente pequeños usando coordenadas horizontales, que a continuación transformaremos al sistema ecuatorial, para finalmente aplicarles la misma proyección que a los elementos del disco fijo.

En este proceso, los segmentos se calculan trivialmente en el sistema horizontal haciendo barridos de

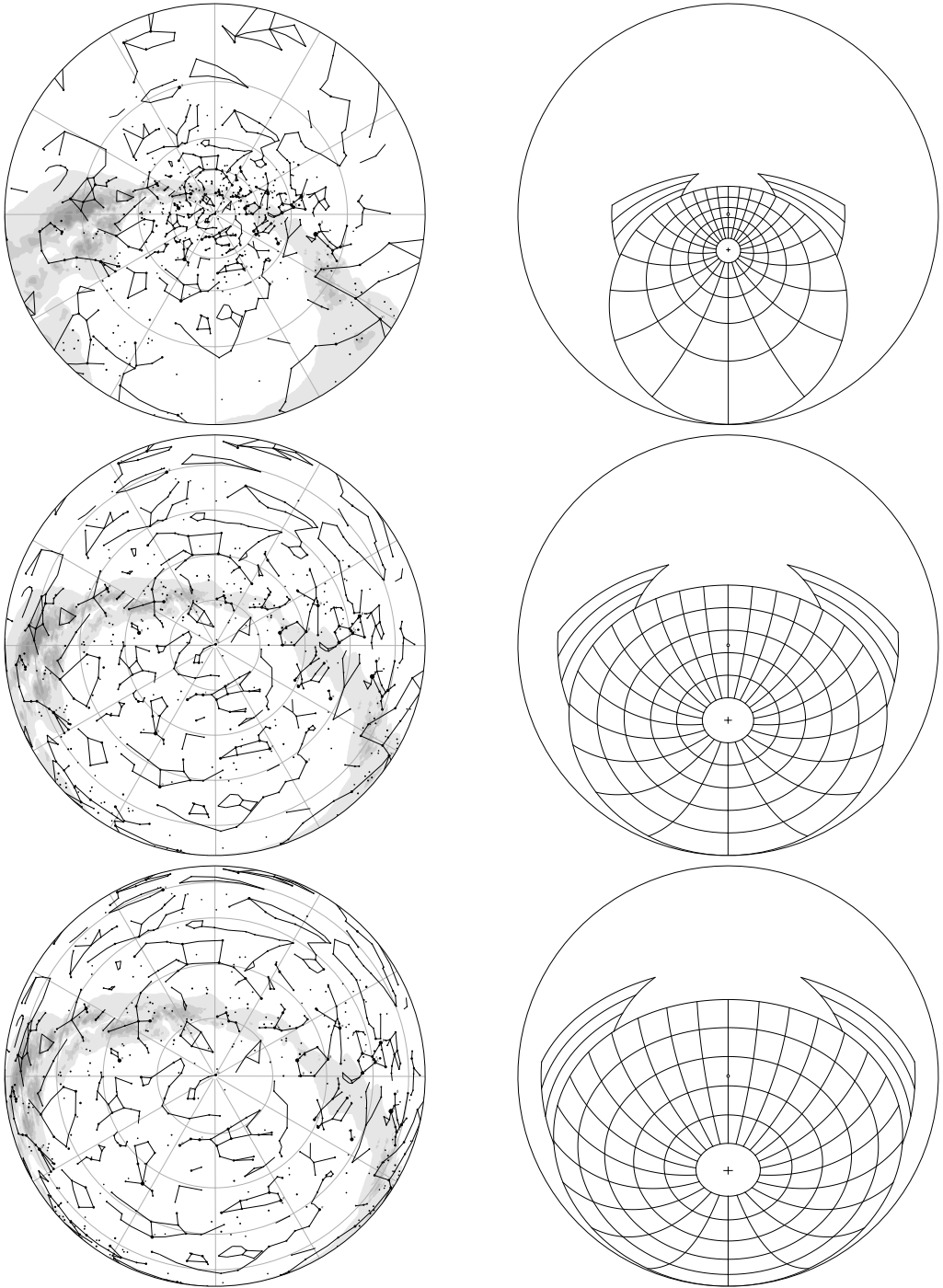


Figura 6. Comparativa de distintas proyecciones: de arriba a abajo, estereográfica, acimutal equidistante y acimutal de áreas equivalentes de Lambert. A la izquierda, los discos fijos, y a la derecha, las ventanas de horizonte y crepúsculos, correspondientes a una latitud de 40°N .

acimut o altura constante. Para la transformación de coordenadas horizontales a ecuatoriales, consideramos una ubicación geográfica sobre el meridiano de Greenwich, con latitud φ , y una orientación de referencia en la que el equinoccio vernal esté culminando. En esta configuración, el polo norte celeste y el equinoccio vernal quedan en el mismo plano que el polo norte geográfico y el cenit, bastando una rotación de $90^\circ - \varphi$ alrededor del vector normal a dicho plano para transformar de un sistema al otro.

En la práctica, tal como se refleja en el *notebook* [1], no será necesario implementar esta transformación, ya que *cartopy* nos permite definir un sistema de referencia cuyo polo local apunte en la dirección del cenit del lugar, lo que a todos los efectos nos servirá para especificar las coordenadas horizontales durante el trazado de la ventanas de horizonte y crepúsculos. *cartopy* realiza de forma automática la transformación al sistema ecuatorial y aplica la misma proyección cartográfica usada para la representación gráfica de ambos discos.

8.4. Rango de acimuts solar para una altura dada

Para poder trazar las ventanas adicionales relativas a los crepúsculos, es necesario calcular el rango de acimuts solar para las alturas $h = \{0^\circ, -6^\circ, -12^\circ, -18^\circ\}$. Los valores resultantes delimitarán las "paredes" laterales de dichas ventanas.

Partimos de las ecuaciones de transformación de coordenadas horizontales en ecuatoriales, en particular de aquella que nos proporciona la declinación, δ :

$$\sin \delta = \sin \varphi \sin h + \cos \varphi \cos h \cos A$$

donde φ es la latitud y (A, h) las coordenadas horizontales, acimut (con origen en el punto cardinal norte) y altura, respectivamente. Despejando el término en acimut, $\cos A$:

$$\cos A = \frac{\sin \delta - \sin \varphi \sin h}{\cos \varphi \cos h}$$

La declinación solar varía en el rango $-\epsilon \dots \epsilon$, donde ϵ es la oblicuidad terrestre. Tomando $\epsilon = 23.44^\circ$, valor actual aproximado de oblicuidad media, podemos usar la expresión anterior para determinar el rango de acimuts solar correspondiente a una altura h dada, considerando $\delta = \{-\epsilon, \epsilon\}$. Al aplicar la función \cos^{-1} para obtener A , las soluciones positivas nos darán el rango de acimuts para el orto, mientras que las negativas, simétricas, corresponderán a las del ocaso. Dependiendo de la latitud φ y de la altura h considerada, es posible que la expresión para $\cos A$ evalúe fuera del rango $[-1, 1]$: esto nos indica que el Sol no atraviesa el almicantrat h .

8.5. Calibración del círculo de fechas

Posiblemente el aspecto más crucial en el diseño del planisferio lo constituya la correcta colocación de las fechas.

En primer lugar debemos notar que en el disco fijo, la ascensión recta (expresada en horas para constituir el círculo horario), crece en sentido de las agujas del reloj para el hemisferio norte, e inverso para el sur. Debido al movimiento de traslación de la Tierra, de forma aparente los astros adelantan su culminación según transcurren las fechas del año, a razón de 1 hora aproximadamente cada 15 días. Por ello, el círculo de fechas debe crecer en sentido opuesto al círculo horario. De esta forma, una misma posición del disco giratorio debe hacer concordar simultáneamente, p.ej. el día 1 de enero con las 3^h y el día 15 de enero con las 2^h .

Por otra parte, es necesario establecer un ángulo origen para el círculo horario que haga que la ventana quede en la posición correcta respecto al disco fijo, aspecto al que nos referimos como "calibración" en

este apartado. Expresado en otros términos, se trata de determinar en qué posición angular del círculo de fechas debemos colocar una determinada fecha; basta con calcularlo para una cualquiera, y el resto de fechas del año quedarán distribuidas uniformemente por todo el círculo. Para determinar la calibración debemos recurrir a una expresión que describa la rotación de la Tierra, relacionando el tiempo con la posición de las estrellas fijas en el cielo para una ubicación de referencia.

La hora sidérea es la medida del ángulo horario del equinoccio vernal o punto Aries, que se mide desde el meridiano local. Tomando como referencia el meridiano de Greenwich, y considerando el equinoccio medio (es decir, teniendo en cuenta la precesión de los equinoccios pero no la nutación), la hora sidérea media en Greenwich, GMST (*Greenwich Mean Sidereal Time*) a las 0^h UT1 viene dada por [13]:

$$GMST(0^h UT1) = \text{mod}(6.697375 + 0.065709824279(JD - JD_{2000.0}), 24)^h$$

donde *mod* es el operador módulo, JD es la fecha juliana UT1 y $JD_{2000.0}$ es la fecha juliana correspondiente al 1 de enero de 2000 a las 12:00 UT1. Esta expresión ignora los términos de segundo y mayor orden ya que no son necesarios para el planisferio.

Haciendo $GMST(0^h UT1) = 0^h, 24^h, 48^h, \dots$ en la ecuación anterior podemos determinar para qué conjunto de fechas julianas el equinoccio vernal está culminando para cualquier lugar del meridiano de Greenwich. En particular, si hacemos $GMST(0^h UT1) = 24^h$ y despejamos JD en la ecuación anterior, obtenemos $JD = 2451808.3187$. Dado que la expresión determina el valor de GMST a las 0^h , debemos redondear el resultado anterior a la medianoche más cercana. Es decir, $JD(0^h) = 2451808.5$, que se corresponde con la fecha gregoriana 21/09/2000, con una desviación de 42.89^s :

$$GMST(0^h UT1) = \text{mod}(6.697375 + 0.065709824279(JD(0^h) - JD_{2000.0}), 24) = 0.0119^h = 42.89^s$$

Si evaluamos esta desviación para el 21 de septiembre entre los años 2000 y 2050, obtendremos que no supera los 2 minutos, siendo en promedio inferior a 2s. Por tanto, queda dentro de la precisión del planisferio, que ronda los 5 minutos.

Si el planisferio está diseñado para cualquier lugar ubicado a lo largo del meridiano de Greenwich, esta fecha de referencia, el 21 de septiembre a las 0:00, es la que debe corresponder a la posición del meridiano local de la ventana, Fig. 7. De esta forma, cuando situemos el meridiano local sobre las 0^h del disco fijo, haremos culminar al equinoccio vernal.

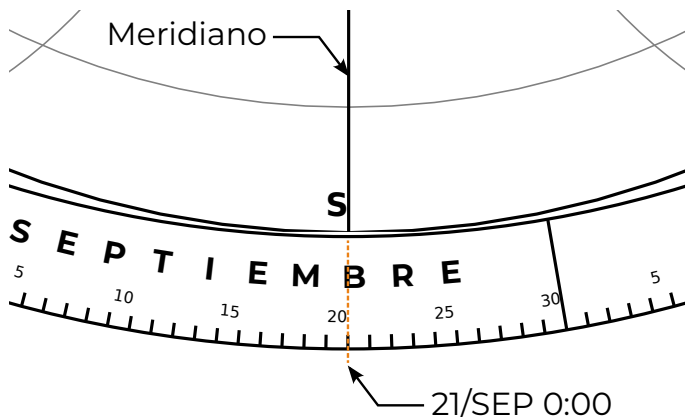


Figura 7. Calibración del círculo de fechas.

Si queremos incluir la compensación correspondiente a la longitud geográfica del lugar de observación así como el desfase debido a su zona horaria, lo haremos corrigiendo la posición angular de la fecha de referencia con un ángulo dado por la expresión:

$$\Delta\tau = 2\pi \frac{\Delta Z - \lambda/15}{24} \left(1 + \frac{1}{365} \right)$$

donde $\Delta\tau$ es la corrección angular en radianes, ΔZ es el desfase correspondiente a la zona horaria (p.ej, +1 en caso de zona UTC+1) y λ es la longitud geográfica, expresada en grados sexagesimales. El término $\frac{1}{365}$ incluye el efecto de la compensación horaria sobre la propia escala de fechas.

Para el hemisferio norte, la corrección del desfase se aplicará restando $\Delta\tau$ a la posición de referencia; para el hemisferio sur, se aplicará sumando.

9. Conclusiones

En el desarrollo del curso de cálculo astronómico impartido por la FAAE² se propuso como ejercicio la implementación de un planisferio celeste. La aplicación web presentada aquí es la evolución de ese proyecto.

En su faceta de ejercicio del curso, ha supuesto un reto en cuanto a manejo de proyecciones cartográficas, cálculo de los crepúsculos y calibración de fechas y desfases horarios. Este artículo y el *notebook* [1] suplementario se han centrado en estos aspectos dado el ámbito de la revista. Gran parte del esfuerzo se ha invertido en la programación del acabado gráfico, que ha conllevado una proporción considerable de líneas de código. La adaptación necesaria para poder ofrecer la generación del planisferio como aplicación web no es compleja pero ha requerido un análisis y mejora del rendimiento para reducir el tiempo de respuesta.

Como herramienta de aprendizaje para talleres de asociaciones astronómicas, enseñanza primaria y secundaria, etc., el planisferio FAAE viene acompañado por una guía didáctica [3] que facilita el diseño de actividades en torno al mismo, conectando los distintos elementos del planisferio con *semillas didácticas* de astronomía: movimiento diurno y estacional, constelaciones y asterismos, coordenadas ecuatoriales y horizontales, orto, culminación y ocaso de astros, entre otras.

Por último, los planisferios generados con la aplicación tienen la vocación de servir como herramienta útil a la comunidad de la astronomía *amateur*, tanto a nivel principiante como experimentado, especialmente en observaciones en campo, donde podemos tener limitaciones en el acceso a dispositivos electrónicos, ya sea por conservar la adaptación visual a la oscuridad o por insuficiente nivel de carga de la batería. Su bajo coste de producción así como la posibilidad de personalización geográfica facilitan su disponibilidad y sencillez de uso.

Referencias

- [1] S. Díaz. *Notebook Jupyter Planisferio FAAE*. https://github.com/JCAAC-FAAE/No02-Mar2025-El_planisferio_FAAE
- [2] S. Díaz. El planisferio FAAE: aplicación web. https://sergiodiaz.eu/planisferio_faae/
- [3] S. Díaz. El planisferio FAAE: guía didáctica. Proyecto del programa Divulgación FAAE. 2024. https://sergiodiaz.eu/planisferio_faae/planisferio_faae_guia_didactica.pdf
- [4] E. Velasco, P. Velasco. Planisferio Procivel. Ed. Procivel. 2002.
- [5] V. Agafonkin. Leaflet: a JavaScript library for interactive maps. <https://leafletjs.com/>
- [6] Yale Bright Star Catalog. <http://tdc-www.harvard.edu/catalogs/bsc5.html>

²El material del curso está disponible en <https://federacionastronomica.es/index.php/group-ca/cursos>

- [7] D. Nash. Augmented Tycho - HYG (AT-HYG). <https://github.com/astronexus/ATHYG-Database>
- [8] D. Nash. HYG Database. <https://github.com/astronexus/HYG-Database/tree/main/misc>
- [9] G. Zotti *et al.* Stellarium. Modern Sky Culture. <https://github.com/Stellarium/stellarium/blob/master/skycultures/modern/constellationship.fab>
- [10] O. Frohn, D. Hernangómez, S. Díaz. Milky Way GeoJSON. https://github.com/sergio-dr/mw_geojson
- [11] E. Lutz. Western Constellations Atlas of Space. https://github.com/eleanorlutz/western_constellations_atlas_of_space
- [12] UK Met Office. Cartopy: a cartographic python library with a Matplotlib interface. <https://scitools.org.uk/cartopy>
- [13] USNO Astronomical Applications Department. Computing Approximate Sidereal Time. <https://aa.usno.navy.mil/faq/GAST>

APPLICATIONS

Asteroide Apophis, *el destructor*

Juan Carlos Terradillos¹¹Aula de Astronomía de Fuenlabrada, Madrid, Spain. E-mail: jcterradillos@gmail.com.**Keywords:** Apophis, asteroides, órbitas

© Este artículo está protegido bajo una licencia Creative Commons Attribution 4.0 License

Resumen

Este artículo aborda el asteroide (99942) Apophis, descubierto en 2004, cuyo potencial riesgo de colisión con la Tierra ha captado la atención de la comunidad científica desde entonces. Su próximo acercamiento, el 13 de abril de 2029, lo llevará a aproximadamente 29000 km de la superficie terrestre, situándolo por debajo de la órbita de los satélites geoestacionarios. Además de describir las características principales de Apophis y contextualizar el peligro que representan asteroides de este tipo para la Tierra, analizamos en detalle los cálculos sobre la distancia mínima a la que pasará y exploramos las misiones espaciales en curso y en desarrollo destinadas a estudiar este objeto potencialmente peligroso.

Abstract

This article discusses the asteroid (99942) Apophis, discovered in 2004, whose potential risk of collision with Earth has captured the attention of the scientific community ever since. Its next approach, on April 13, 2029, will bring it to approximately 29000 km from Earth's surface, placing it below the orbit of geostationary satellites. In addition to describing Apophis' main characteristics and contextualizing the threat posed by such asteroids to Earth, we thoroughly analyze the calculations regarding its minimum passing distance and explore ongoing and developing space missions aimed at studying this potentially hazardous object.

1. Introducción

Con relativa frecuencia uno puede oír en los diferentes medios de comunicación noticias, en ocasiones acompañadas por imágenes reales, que nos informan del impacto sobre la Tierra de cuerpos celestes y de las catastróficas consecuencias que pueden tener para nuestro planeta.

Así mismo abundan los artículos en prensa y revistas especializadas, más o menos sensacionalistas en su línea editorial, que tratan sobre este tema, incidiendo de una manera tendenciosamente morbosa en el riesgo que estos eventos astronómicos suponen para toda la humanidad, sensación que se acrecienta cuando vemos documentales, tan ilustrativos y bien realizados para poder llegar a todo tipo de público en general, que nos causan asombro y nos encogen el corazón por la incertidumbre que añaden sobre nuestra existencia, a la vez que nos presentan de forma cruda lo que, para los astrónomos, es una objetiva y tozuda realidad: este tipo de sucesos han ocurrido siempre en el pasado, siguen dándose en el presente y sucederán en el futuro.

Esta idea nos acompaña a modo de resignación pues, aunque se presenten de vez en cuando de manera aleatoria en el tiempo, no los podemos predecir y, mucho menos, combatir sus terribles efectos, por el momento. ¿Quién no ha visto, cómodamente sentado en su casa delante del televisor, al enorme asteroide que se dirige hacia la Tierra en la película “Armageddon”, del año 1998, o al masivo cometa que impacta sobre la superficie terrestre en la de “Deep Impact”, también de 1998?

Las dos películas fueron hechas el mismo año del siglo pasado, lo cual reflejaba la sensibilidad que en esos años había respecto al Fin del Mundo, consecuencia de las predicciones apocalípticas enunciadas por distintos visionarios al ver cerca el cambio de milenio, sensación colectiva muy parecida a lo ocurrido mil años antes en plena Edad Media. Es por esto que, con el objetivo de tomar conciencia sobre estos peligros provenientes del espacio exterior, la Asamblea General de Naciones Unidas, en 2016, declaró el 30 de junio como Día Internacional de los Asteroides, en recuerdo de que fue ese día de 1908 cuando cayó sobre la región rusa de Siberia, el conocido como “Bólido de Tunguska”.



Figura 1. Recreación del Bólido de Tunguska. Créditos: Don Davis, <http://donaldedavis.com/>.

Yo también he vivido indirectamente la caída de un asteroide sobre la Tierra. Me refiero al llamado “Bólido de Chelyabinsk”, que impactó sobre esa pequeña población rusa el día 15 de febrero de 2013. Esa noche me hallaba en el patio de mi casa, observando con el telescopio la región del cielo correspondiente a la constelación de Casiopea, en busca de los bonitos cúmulos estelares que hay en ella. Sería sobre las diez de la noche cuando vi pasar por el campo del ocular que tenía en ese momento puesto, ¿qué casualidad!, una estela luminosa. Me quedé asombrado, levanté la vista al cielo y pude comprobar como una banda de humo, relativamente ancha y larga, ocupaba esa parte del cielo, y también como,

22

pasado un cierto tiempo, se desplazó hacia zonas celestes más altas de mi horizonte local. Excitado por la visión entré en la cocina de casa, donde en ese momento se hallaban mi hijo y mi mujer y dije: “donde caiga ese asteroide va a causar mucho daño”.

Fueron mis alumnos los que al día siguiente me informaron, al entrar en clase, que había caído un asteroide en Rusia y había ocasionado numerosos destrozos y heridos en la mencionada ciudad rusa. Desde ese momento me encontré motivado por este tipo de sucesos astronómicos, sintiendo la necesidad de profundizar en su estudio, para lo cual nada mejor que dedicar mis esfuerzos, conocimientos y tiempo en la elaboración del presente artículo sobre el asteroide Apophis, “El Destructor”.

Resultaba inevitable que dada mi formación como historiador, especialidad en Prehistoria, quisiera abordar la cuestión de ampliar la investigación a otras épocas del pasado, en un intento por buscar, en este tipo de acontecimientos celestes, una posible relación causal para la desaparición, muchas veces inexplicable para la ciencia y repentina en el tiempo, de algunas de las principales culturas históricas o prehistóricas que existieron sobre nuestro planeta. Es muy ilustrativo al respecto la cantidad de evidencias científicas que nos hablan de la extinción masiva de animales en otras épocas geológicas lejanas y, más recientemente, durante el Holoceno, en el Tardiglacial o Dryas Reciente (10900 – 9700 a.C.). Igualmente, algunos de los mitos asentados en la conciencia de todos los hombres, como pudo ser el Diluvio Universal, o en particular en el espacio circunmediterráneo clásico, como el mito de la Atlántida o la destrucción bíblica de las ciudades de Sodoma y Gomorra, pueden tener una explicación relacionada con el impacto de un cuerpo celeste proveniente del espacio exterior.

Asumido este riesgo real que corre toda la humanidad, era cuestión de tiempo que las instituciones, los estados y las distintas Agencias Espaciales de los países desarrollados, tomaran la decisión de avanzar en la identificación y el seguimiento de estos peligrosos objetos que se acercan a la Tierra, llamados NEOs (Near Earth Object), y en idear una serie de métodos que pudieran resolver el problema de un posible impacto, pues va en ello nuestra propia supervivencia e, incluso, de la vida en general como la conocemos ahora.



Figura 2. Bólido de Chelyabinsk [1].

Con el descubrimiento en 2004 de Apophis y de las predicciones funestas que sobre él se hicieron en ese momento, se dio el primer paso para que los astrónomos y científicos reflexionaran sobre la manera de interceptar, hacer variar su órbita o destruir estos NEOs, programando misiones que pudieran cumplir este objetivo.

A día de hoy, ya se han logrado alcanzar unos resultados notables con este tipo de misiones espaciales, muy laboriosas y extensas en el tiempo de planificación y ejecución de las mismas, habiéndose conseguido ya tomar contacto en la superficie de algunos asteroides, recoger y enviar muestras a la Tierra para su estudio, así como impactar y explosionar una carga detonante que alterara la trayectoria seguida en el espacio por estas enormes rocas que viajan a grandes velocidades. En concreto, ya está en marcha una misión específica para nuestro asteroide Apophis aprovechando su aproximación a la Tierra a una distancia de unos pocos radios terrestres, que tendrá lugar en la primavera de 2029.

Realizada esta introducción sobre el tema a tratar en el presente artículo, se hace necesario, antes de abordar el desarrollo de los distintos apartados del mismo, recoger los objetivos a conseguir con este trabajo.

1. Conocer las características físicas y dinámicas del asteroide.
2. Presentar un método de cálculo geométrico para poder hallar la Mínima Distancia Orbital (MOID) entre las órbitas de la Tierra y de Apophis en el espacio.
3. Calcular la distancia (en km) a la que se hallará Apophis en su momento de acercamiento máximo a la Tierra, en abril del año 2029.
4. Calcular las Coordenadas Eclípticas Heliocéntricas de Apophis, respecto al equinoccio medio, en la fecha de estudio.
5. Calcular sus Coordenadas Eclípticas Geocéntricas, en el marco de referencia J2000.
6. Calcular sus Coordenadas Ecuatoriales J2000.0 y las Horizontales locales para un observador situado en la ciudad de Madrid, en el momento de estudio.
7. Visualizar gráficamente su posición en el cielo durante su acercamiento máximo.

Junto a todos los anteriores objetivos planteados, quiero añadir un objetivo general que está presente en todos mis trabajos astronómicos, que consiste en exponer, de la manera más didáctica posible todos los pasos seguidos, para dotar así de un soporte matemático el desarrollo de los contenidos específicos de cálculo presentes en el mismo. Dada mi situación actual de docente no activo -por jubilación-, se entenderá que mi intención pedagógica es poder ofrecer un modelo a seguir para los lectores que se formulen, en un momento dado, unos mismos o parecidos objetivos a conseguir, respecto a otros cuerpos celestes del Sistema Solar, ya sean NEOs, cometas o planetas, que sigan órbitas elípticas en el espacio.

Finalmente, quiero expresar un deseo personal en esta introducción. Consiste en la intención que manifiesto ahora para que este trabajo sea el primero de otros más que estarían por llegar, que trataran sobre otros escenarios de impacto de cuerpos celestes sobre la Tierra, que tuvieron un resultado catastrófico sobre algunas regiones geográficas de la misma y sobre algunas culturas y sociedades, en un pasado reciente o más alejado en el tiempo. La Rusia revolucionaria de principios del siglo XX, junto con mis recientes visitas turísticas al Mar Muerto y al yacimiento turco de Gobekli Tepe, pueden dar alguna pista sobre este deseo investigador.

2. El asteroide 99942 Apophis

Apophis es una enorme roca espacial, de unos 325 m de diámetro, con forma irregular bilobulada, y cierto parecido a un cacahuete, según observaciones de radar realizadas en 2021. Su color es bastante oscuro, con un albedo de 0.23 –correspondiendo al nivel 1 un cuerpo totalmente reflectante por lo que será visible a simple vista durante su aproximación a la Tierra de 2029–. Su masa es aproximadamente unas 21×10^6 Tm, correspondiéndole una densidad media de 3200 kg/m^3 .

En cuanto a su composición material, contiene principalmente silicatos de hierro y magnesio, presentes en rocas ultrabásicas de olivino y piroxeno, en distintas proporciones porcentuales, con una génesis ígnea y metamórfica en su formación natural. Al igual que ocurre con todos los asteroides, Apophis es un remanente material del periodo de formación de nuestro Sistema Solar, que tuvo lugar hace unos



Figura 3. Tamaño y forma de Apophis [2].

4600 millones de años. En un principio se hallaba formando parte del Cinturón de Asteroides, que se encuentra orbitando alrededor del Sol, entre las órbitas de Marte y Júpiter. Con el transcurso de millones de años, su órbita fue cambiando, afectada por las perturbaciones gravitacionales que actuaban sobre él procedentes de los grandes planetas, principalmente de Júpiter. Fue de este modo como se convirtió en un NEO.

En un principio, cuando se descubrió en el año 2004, se le dio el nombre de 2004 MN₄, para cambiar en el 2005 a su denominación definitiva: asteroide 99942, añadiéndole el sobrenombre de Apophis ese mismo año. Este nombre fue elegido por sus descubridores R. A. Tucker, D.J. Tholen y F. Bernardi, investigadores en el Observatorio Kitt Peak de Arizona, no tanto por resaltar la destrucción que en la mitología egipcia simbolizaba el dios Apep, como por su condición de seguidores de la serie de ciencia ficción de Stargate, muy popular en esos años.

Para los egipcios, Apophis o Apep era sinónimo de maldad, asociado al Duat, encarnación del inframundo, en donde tenía lugar el Juicio para la salvación del alma del difunto, en la presencia del dios Osiris. En las pinturas murales de algunas tumbas se puede ver a Apep representado como una enorme y poderosa serpiente. Apophis es un asteroide difícil de observar, porque es visible por la mañana o por la tarde, siempre a poca distancia angular del Sol, durante su perihelio, cuando su distancia a la estrella es mínima. Esta dificultad ha provocado que fuera más complicado el seguimiento y posterior cálculo de sus elementos orbitales con exactitud, siendo esta la causa de las especulaciones catastrofísticas de impacto contra la Tierra en el futuro. Nuevas observaciones visuales y mediciones realizadas con radar han descartado esta posibilidad para todo el siglo XXI, aunque existen ciertas incertidumbres respecto a algunos años, enunciadas por alguna Agencia Espacial, como ocurre en el año 2068, derivadas de la alteración de su órbita ocasionada por el efecto Jarkovsky, un fenómeno físico relacionado con la manera como los asteroides absorben la luz del Sol y la irradian después al espacio, para mantener su equilibrio térmico, dando lugar a una aceleración en el cuerpo celeste.

Se entiende pues que, en un principio, el asteroide figurara en la Escala de Turín con el nivel 1, es decir contemplándose ciertas posibilidades de impacto contra la Tierra. En esta escala, un NEO catalogado en el nivel 0 significa una posibilidad casi nula de impacto, mientras que el nivel 10 nos indicaría un choque seguro y, por consiguiente, unas consecuencias planetarias devastadoras sobre amplias zonas de la misma.



Figura 4. Escena de la tumba de Ramsés IV con la serpiente del dios Apep [3].

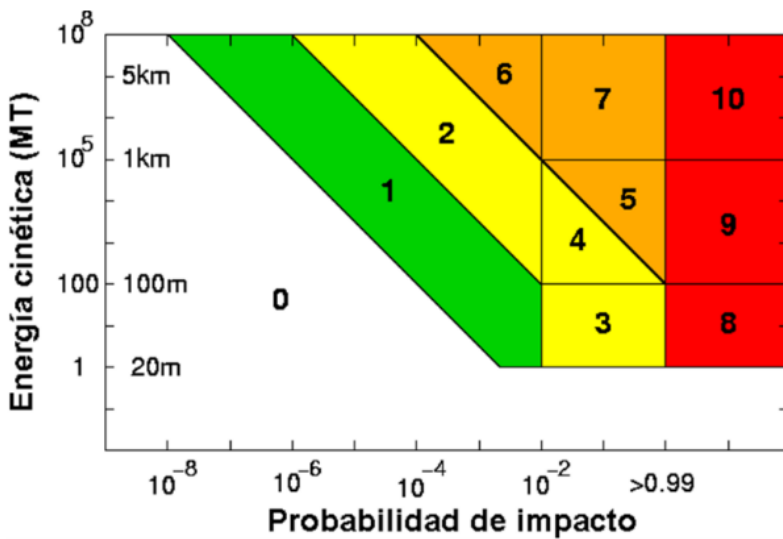


Figura 5. Escala de Turín [4].

3. La órbita de Apophis

Sus características orbitales son las correspondientes a un asteroide tipo Atón, asteroides con órbitas cuyo semieje mayor (a) es menor que el de la Tierra (1 UA= 149597870 km). Tiene poca excentricidad (e), por lo que su órbita no se aleja mucho de una circunferencia. No obstante, cruza dos veces la órbita terrestre durante su trayectoria periódica alrededor del Sol, que dura un poco menos de un año, desplazándose en el espacio en el mismo sentido que lo hace nuestro planeta. El periodo de rotación sobre su eje axial es de 30.56 horas. Una consecuencia de su máxima aproximación a la Tierra del

año 2029, será que su órbita se hará más ancha y grande que 1 UA, pasando entonces a formar parte de una nueva categoría de asteroides, llamados del tipo Apolo. En la Fig. 6 se presenta un cuadro en donde se recogen los elementos orbitales y otras medidas angulares y de tiempos, correspondientes a tres momentos distintos.

Elementos orbitales	13 abril de 2028 a las 21 h 46m (TDB)	13 abril de 2029 a las 21 h 46m (TDB)	13 abril de 2036 a las 21 h 46m (TDB)
e	0,1911193205	0,2236577328	0,189163609605
Ω°	203,8789943779	203,7588103533	203,51842547804
n°	1,1125050926	0,9592261242	0,85080497853
a (U.A.)	0,9224327437	1,0182522627	1,103018257063
q (U.A.)	0,746138024554	0,7905122701	0,894367342096
ω°	126,7247597701	99,2041738541	71,483435215251
M°	206,25322400887	286,6376440891	323,90859832898
Q (U.A.)	1,0987274630048	1,2459922553	1,311669172029
i°	3,3412271204066	3,6256376945	2,2215090354122
T_p (DJ)	2462013,605661	2462316,887717	2464839,82724355
V°	198,30809815473	260,8589107495	308,35631297100
P (días)	323,59402431609	375,3025391172	423,1286946843

Figura 6. Elementos orbitales de Apophis correspondientes los máximos acercamientos a la Tierra en los años 2028, 2029 y 2036. Los elementos orbitales son: e , excentricidad; Ω , longitud del nodo ascendente (en grados); n , movimiento medio diario (en grados por día); a , semieje mayor (en UA); q , distancia en el perihelio (en UA); ω , argumento de latitud del perihelio (en grados); M , anomalía media (en grados); Q , distancia en el afelio (en UA); i , inclinación (en grados); T_p , instante de paso por el perihelio (DJ); V , anomalía verdadera (en grados), y P , periodo (en días). Fuente: JPL Horizons online solar system data and ephemeris.

Primero, los referidos al año 2028, después los pertenecientes a su mayor acercamiento a la Tierra en el mes de abril de 2029 y también los del año 2036, otro paso en el que se acercará mucho a nosotros y que originó, en su momento, incertidumbre sobre un posible impacto contra nuestro planeta, afortunadamente ya descartado por NASA.

Observando detenidamente algunos elementos orbitales, vemos como el campo gravitatorio de la Tierra hace que la excentricidad (e), el semieje mayor (a) y, por tanto su perihelio (q) y su afelio (Q), además de la inclinación (i) y el periodo (P) aumenten sus cantidades de medida, reflejando de este modo como una nueva órbita ha resultado de su acercamiento extremo a nuestro planeta en 2029.

Este cambio acontecido en la órbita, junto con la existencia de una casi resonancia en la proporción 7:6, entre las órbitas de Apophis y la Tierra, es decir, siete traslaciones alrededor del Sol del asteroide son casi igual a seis traslaciones anuales del planeta; harán que su aproximación de 2036, a unas 0.309 UA

de nosotros, sea bastante más lejana que la alcanzada en el año 2029, de 0.00025 UA, aproximadamente unos 38000 km, correspondiente a un poco más de seis radios terrestres o 1/10 parte de la distancia media Tierra-Luna.

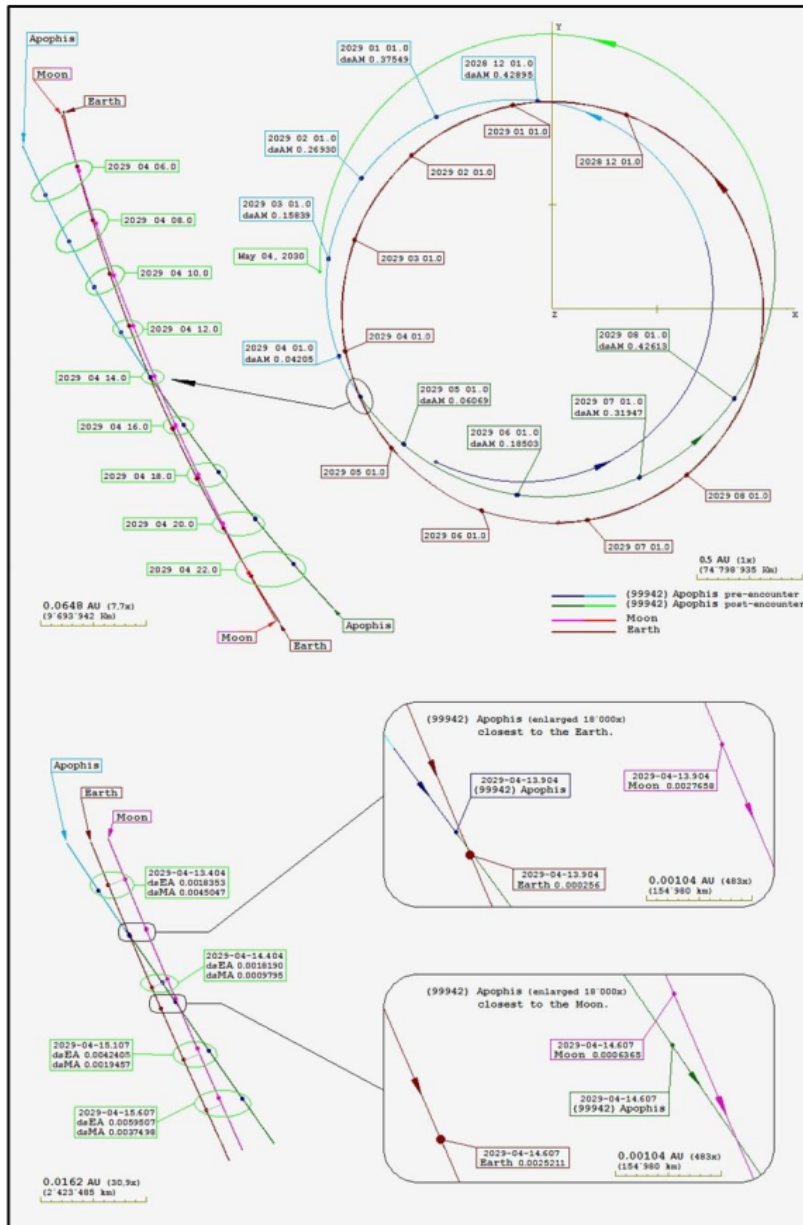


Figura 7. Trayectorias de Apophis, desde el 1 diciembre 2028 hasta 1 agosto de 2029, (vistas desde el polo norte de la eclíptica), durante su máxima aproximación al sistema Tierra-Luna [5].

Los tiempos en los que están medidos los distintos elementos orbitales osculadores (elementos instantáneos de la órbita) corresponden a Tiempo Dinámico Baricéntrico (TDB), un tiempo referido al centro

28

de masa del Sistema Solar que contempla correcciones relativistas. Difiere ligeramente del Tiempo Terrestre (TT), en un máximo de 0.0017 segundos derivados del movimiento de la Tierra alrededor del Sol. En adelante, el TDB será en este trabajo igual a TT, una escala de tiempo que tiene en cuenta la rotación de la Tierra, que varía con el paso del tiempo. Existe una relación entre el TT y el Tiempo Universal (UT¹), que corresponde al Tiempo Civil de Greenwich, necesario para el cómputo temporal en nuestra vida civil y para los cálculos astronómicos relacionados con el Ángulo Horario (A.H.), una coordenada horizontal del lugar de observación.

La corrección a aplicar al TT para expresarlo en UT es ΔT . Este incremento varía según el año, de tal manera que $\Delta T = +65s$ en el año 2000, $\Delta T = +69s$ en 2005, y $\Delta T = +80s$ en 2015. La relación matemática es $\Delta T = TT - UT$. Si se quiere hallar este ΔT , de una forma aproximada, para un año concreto, en el pasado o en el futuro, se hace de la siguiente manera²:

Sea t el tiempo medido en siglos desde el año 2000 (t será negativo antes de ese año y positivo después de ella), $t = (\text{año} - 2000)/100$. Entonces, para antes del año +948, $\Delta T = 2177 + 497t + 44.1t^2$, mientras que para los años comprendidos entre el +948 y el +1600 y después del año 2000, $\Delta T = 102 + 102t + 25.3t^2$. Para evitar una discontinuidad con 2000 AD 2000 (After Domine) es necesario añadir una corrección para los años comprendidos entre el 2000 y el 2100 igual a $+0.37(\text{año} - 2100)$. Para el valor de ΔT (en segundos) entre los años 1620 y 2010, ver Material Suplementario 1.

4. Mínima distancia orbital de Apophis (MOID)

Se define esta distancia (*Minimum Orbit Intersection Distance*, en inglés) como la mínima distancia que hay entre dos puntos en las órbitas de dos cuerpos que comparten un mismo foco, siendo el Sol el foco y Apophis y la Tierra los dos cuerpos con órbitas elípticas en el espacio. Estos MOID son importantes de calcular cuando se trata de asteroides peligrosos para la Tierra (como los NEOs), formando parte de los datos con los que trabajan las Agencias Espaciales, para poder valorar el riesgo de colisión entre ellos. No sólo se halla el MOID de los asteroides y cometas, sino también los correspondientes a los planetas del Sistema Solar, para poder elaborar sus efemérides y programar las distintas misiones espaciales hacia todos estos cuerpos celestes. El MOID es la mínima distancia teórica posible entre las órbitas de los dos cuerpos, mientras que la distancia mínima real que alcanzan ambos cuerpos, en un momento dado, se halla en función de las posiciones relativas concretas alcanzadas por ellos en el espacio en ese momento concreto y puntual.

Los pasos a dar para hallar el MOID entre las dos órbitas son los siguientes:

1. Tomar los elementos orbitales de los dos cuerpos celestes, referidos a la misma época y mismo momento de tiempo. Estos elementos son el semieje mayor (a), la excentricidad (e), la inclinación (i), la longitud del nodo ascendente (Ω) y el argumento de perihelio (ω). Si una de las órbitas es la de la Tierra hay que dar sus elementos para esa época y ese mismo día y momento. Los elementos orbitales terrestres son la longitud media (L), semieje mayor (a), excentricidad (e), inclinación (i) y la longitud de perihelio ($\tilde{\omega}$). Tanto a como i tienen un valor constante, mientras que L , Ω y $\tilde{\omega}$ están en función del tiempo (T) transcurrido desde la época J2000.0 (1 de enero a las 12 horas = JD2451545.0), expresado en siglos de 36525 días cada uno.
2. Ya que el método elegido para hallar el MOID es el método simple o geométrico, mucho más sencillo de aplicar que aquellos que utilizan cálculos numéricos y algebraicos, se van a realizar sucesivas aproximaciones en los valores de MOID, mediante iteraciones en los valores introducidos,

¹Denotamos por UT a la versión del Tiempo Universal llamada UT1.

²Valores más precisos de ΔT pueden encontrarse en forma tabular en <https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html>

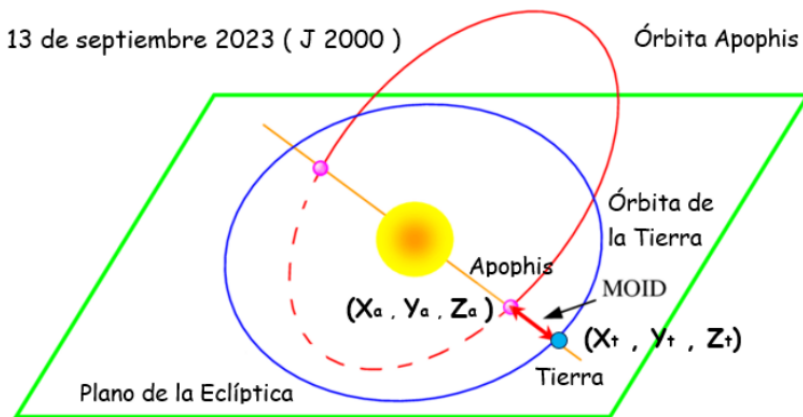


Figura 8. *Mínima Distancia Orbital Apophis-Tierra.*

hasta alcanzar finalmente el objetivo deseado: encontrar el MOID entre Apophis y la Tierra, con el suficiente grado de exactitud. Este método tiene varias fases en su desarrollo.

- Hallar las Coordenadas Eclípticas Heliocéntricas (X, Y, Z) de cada uno de los cuerpos celestes cuyo MOID se quiere calcular, así como sus respectivas distancias al Sol, para cada una de las longitudes eclípticas elegidas de ambos objetos celestes (λ_a, λ_t) , de Apophis y Tierra, respectivamente, dándoles igual valor (en grados) a ambas longitudes, pues suponemos que el MOID se encontrará en el entorno de alguno de los nodos de la órbita del asteroide. Estas coordenadas serán X_a, Y_a, Z_a , para las de Apophis, y X_t, Y_t, Z_t para las de la Tierra; mientras que sus distancias respectivas al Sol serán r_a y r_t .
- Calcular la distancia (Δ) que hay en el espacio entre los dos cuerpos, para cada par de las longitudes eclípticas orbitales, λ_a, λ_t (de igual valor para las dos), elegidas anteriormente.
- Localizar dónde se produce el valor mínimo de estas distancias (Δ) y ver qué valores de λ_a, λ_t les corresponde. Ya tenemos el entorno inmediato de las longitudes eclípticas orbitales de ambos cuerpos. Tomando como referencia este entorno, probar variando ligeramente estas longitudes orbitales de Apophis y de la Tierra, qué nuevas distancias Δ con valores más pequeños vamos obteniendo, hasta poder elegir la mínima distancia entre todas ellas. El objetivo en este paso es ajustar, lo máximo posible, el MOID entre ambos cuerpos.
- Una vez obtenida esta distancia mínima Δ obtenida anteriormente, es el momento de aplicar el método simple geométrico para encontrar qué longitudes (más exactas y concretas) corresponden a una distancia Δ que tenga una precisión de 10^{-6} U.A. (ya que $(10^{-6}$ UA = 0.000001 UA = 149.6 km), considerado suficiente como margen de error en Δ .

Este método simple parte de tres pares de valores (λ_a, λ_t) , a modo de vértices de un triángulo, (A, B, C) , para ir luego probando mediante las diferentes estrategias llamadas de *reflexión, expansión, contracción*, y el *encogimiento o reducción*, llevadas a cabo sobre estos pares de longitudes eclípticas orbitales, buscando obtener siempre los mejores valores de Δ alcanzados para cada terna de (λ_a, λ_t) ; es decir, añadiendo como nuevo vértice del triángulo el que tiene mejor valor de la distancia Δ , sustituyéndole por el vértice que tenía el peor valor de los tres pares (λ_a, λ_t) , y que correspondía al valor más alto del anterior triángulo ABC.

Mediante este método geométrico llegamos, finalmente, a conseguir un triángulo final (A, B, C) , en el que todas las distancias Δ logradas para los pares (λ_a, λ_t) , no difieran en ese error de 10^{-6} UA. Decir

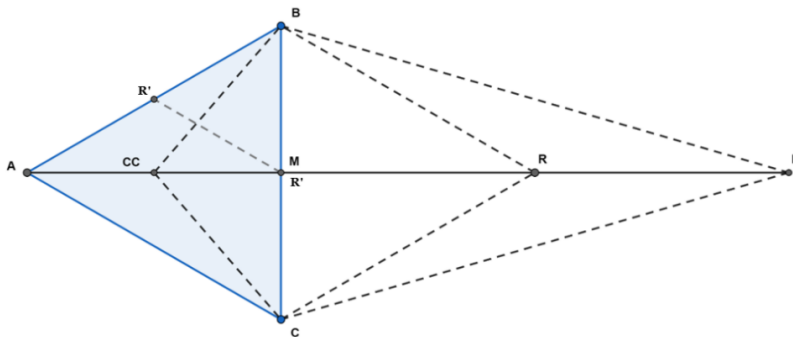


Figura 9. Triángulo geométrico con las distintas acciones a realizar para hallar el MOID. Segmentos en el triángulo: $AM = d$, $MR = RE = d$, $ME = 2d$, $Acc = AM/2$, $AR' = AB/2$, $BM = BC/2$, $BR' = R'C = BC/2$, $M = R'$. Puntos en el triángulo: $M = Par(\lambda_a, \lambda_t)$, medio en el lado BC del triángulo; $R = Par(\lambda_a, \lambda_t)$, reflejado del par A; $E = Par(\lambda_a, \lambda_t)$, expandido del par A; $CC = Par(\lambda_a, \lambda_t)$, medio entre los pares A y M; $R' = Pares(\lambda_a, \lambda_t)$, reducidos desde los pares A y C, hacia el par B.

que, tanto el paso 1 como el 2 -con sus distintas fases- han sido expuestos en términos generales de aplicación, para hacernos una idea general del cálculo del MOID Apophis-Tierra.

Será en el siguiente apartado de este trabajo cuando se describirá, de forma detallada y un poco más gráfica, todos los pasos anteriores seguidos para hallar el MOID entre ambos cuerpos celestes.

5. Cálculo del MOID Apophis – Tierra

Para realizar este cálculo es necesario utilizar los elementos orbitales de Apophis y la Tierra, correspondientes a la misma época, que define en el espacio un marco tridimensional concreto para sus coordenadas X, Y, Z (por ejemplo, J2000.0) y al mismo momento de observación, o de estudio. Esto se justifica porque la precesión de los equinoccios hace variar lentamente el eje X , dirigido hacia el Punto Vernal Υ , el eje Y es siempre perpendicular al X y se encuentra en el mismo plano, mientras que el eje Z es perpendicular al plano XY en el espacio.

Esta fecha o momento temporal de estudio, no ha sido elegida arbitrariamente, sino condicionada por la existencia de unos elementos orbitales de Apophis dados por el Jet Propulsion Laboratory (JPL), en base a las observaciones llevadas a cabo sobre su órbita, desde su descubrimiento en 2004 hasta las realizadas en el 2021. La fecha elegida es el 13 de septiembre de 2023 a las 0h TDB, correspondiente a las 23h y 57m T.U del 12 de septiembre de 2023, ya que $\Delta T = +136$ s. La época de referencia es J2000, correspondiente al 1 de enero de 2000, a las 12h.

1. **Elementos orbitales de Apophis (J2000.0).** Los elementos orbitales del asteroide ofrecidos en la Fig. 10 se encuentran referenciados respecto al Sol, es decir son heliocéntricas, y respecto al plano de la Eclíptica que es el formado en el espacio por el Sol y la trayectoria seguida por la Tierra en su movimiento de traslación alrededor suyo. Estrictamente hablando, estos elementos orbitales eran los que presentaba Apophis en la fecha dada, el 23 septiembre de 2023 a las 0h TDB (*osculating orbital Elements*), o en el entorno de ella como pueden ser unos pocos meses, pero yo me voy a servir de ellos para mi propósito de hallar el MOID entre las dos órbitas elípticas. Los elementos que aparecen en la misma son: **e**, excentricidad; **a**, semieje mayor; **q**, distancia en el perihelio; **i**, inclinación; **node**, longitud del nodo ascendente; **peri**, argumento de latitud del perihelio; **M**, anomalía media; **tp**, instante de paso por el perihelio; **period**, periodo de traslación, **n**, movimiento medio diario; y **Q**, distancia en el afelio.

Osculating Orbital Elements

Epoch 2460200.5 (2023-Sep-13.0) TDB Reference: JPL 216 (heliocentric IAU76/J2000 ecliptic)			
Element	Value	Uncertainty (1-sigma)	Units
e	0.1914411460533765	1.7677E-9	
a	0.9227218395034338	1.6226E-9	au
q	0.7460749130604167	2.7915E-9	au
i	3.339296377069885	1.104E-7	deg
node	203.9566720818881	3.7763E-6	deg
peri	126.6039880940337	4.089E-6	deg
M	142.8571421049518	6.8509E-7	deg
tp	2460072.029302035401 2023-May-07.52930204	7.0849E-7	TDB
period	323.7461605754216	8.5397E-7	d
	0.8863686805624137	2.3380E-9	y
n	1.111982299219059	2.9331E-9	deg/d
Q	1.099368765946451	1.9332E-9	au

Figura 10. Elementos orbitales de Apophis el 23 de septiembre de 2023, a las 0h TDB.

Tierra	a_0	a_1	a_2	a_3
L (longitud media del planeta)	100,466457	35999,3728565	- 0,00000568	- 0,000000001
i (inclinación respecto eclíptica)	0,0000000	0,0130548	- 0,00000931	- 0,000000034
Ω (longitud del nodo ascendente)	174,873176	- 0,2410908	0,00004262	0,000000001
ϖ (longitud del perihelio)	102,937348	0,3225654	0,00014799	- 0,000000039

Figura 11. Coeficientes para los elementos orbitales de la Tierra con respecto al equinoccio medio J2000.0.

2. **Elementos orbitales de la Tierra.** Si los elementos orbitales de Apophis, en esa fecha, han sido tomados del JPL, los de la Tierra tienen que ser calculados. Todos los elementos orbitales de los planetas del Sistema Solar y, por tanto también los de la Tierra, pueden ser expresados como una función polinómica de la forma:

$$a_0 + a_1T + a_2T^2 + a_3T^3, \quad (5.1)$$

en donde T es el tiempo, medido en siglos de 36.525 días, desde la fecha de estudio hasta la época de referencia J2000.0, igual a JD2451545.0 (Días Julianos). El tiempo T se calcula mediante

$$T = (JD - 2451545.0)/36525. \quad (5.2)$$

En nuestro caso el valor de T es igual a 0.023697467488 siglos. En esta expresión 2460200.5 corresponde al JD del 13 de septiembre de 2023 a las 0h TDB \approx 0h UT. El resultado es dado con suficiente número de decimales ya que las unidades de T son siglos y con esto se gana en precisión en la medida del tiempo. Los coeficientes a_0, a_1, a_2, a_3 para la Tierra y el equinoccio J2000.0 son recogidos en la Tabla de la Fig. 11. Sustituyendo estos coeficientes a_n , y el tiempo T en cada uno de los elementos orbitales de la Tierra, tenemos:

Longitud media de la Tierra: $L_t = 8631.406135^\circ$, que divididos entre 360° suponen 23.9761281534 vueltas, que corresponden a 351.4061352419° . Este resultado nos sitúa a la Tierra sobre el plano de la eclíptica, en la cabeza de la constelación de Piscis, mientras que el Sol se localiza al otro lado de la eclíptica, con una Longitud eclíptica $L_s = 351.4061352419^\circ - 180^\circ = 171.4061352419^\circ$, en la constelación de Virgo en la fecha de estudio (13 de septiembre de 2023, a las 0h TDB).

Procediendo de manera similar para los otros elementos orbitales, tenemos:

Inclinación sobre la Eclíptica: $i_t = 0.0030931337^\circ$
Longitud del nodo ascendente: $\Omega_t = 174.8160459795^\circ$
Longitud del perihelio: $\tilde{\omega} = 103.01379661409^\circ$
Excentricidad: $e_t = 0.0166986612$

Teniendo la longitud del perihelio $\tilde{\omega}$ y la Longitud del nodo ascendente Ω_t , se puede hallar el valor del argumento de latitud del perihelio,

$$\omega_t = \tilde{\omega}_t - \Omega_t, \quad (5.3)$$

y sustituyendo valores se obtiene

$$\omega_t = 103.01379661409^\circ - 174.8160459795^\circ = 288.1977501615^\circ. \quad (5.4)$$

Una vez que se tienen los elementos orbitales de Apophis y de la Tierra, ahora hay que calcular, grado a grado en el intervalo $[0^\circ, 360^\circ]$, sobre cada una de las posibles longitudes eclípticas orbitales, λ_a y λ_t , asignadas (ver punto 2b de la Sección 4), y para cada uno de los cuerpos celestes, los siguientes elementos orbitales:

Argumento de latitud, U , que se define como

$$U = \lambda - \Omega = \text{arco NX} \quad (5.5)$$

Anomalía verdadera, V , que se define como

$$V = \lambda - \Omega - \omega = \text{arco PX} \quad (5.6)$$

Además hay que calcular, en ese mismo intervalo $[0^\circ, 360^\circ]$, el radio vector o la distancia al Sol, tanto de Apophis, r_a , como la de la Tierra, r_t . Una vez que conocemos la excentricidad e_t y la anomalía verdadera de la Tierra V_t para cada grado del intervalo anterior, así como las correspondientes a Apophis (e_a y v_a), haciendo uso de sus elementos orbitales presentes en la Tabla 2, para la fecha de estudio (13 de septiembre de 2023, a las 0h TDB), ya estamos en condiciones de calcular los módulos de los radios vectores, $r_a = |\mathbf{r}_a|$ y $r_t = |\mathbf{r}_t|$, de las distancias al Sol de ambos cuerpos, mediante la fórmula siguiente:

$$r = \frac{a(1 - e^2)}{1 + e \cos V}, \quad (5.7)$$

tomando los valores respectivos. Conocidos los valores de U_a , V_a y r_a en el intervalo, ahora hay que hallar las coordenadas rectangulares eclípticas heliocéntricas del asteroide, X_a , Y_a , Z_a , haciendo uso de las siguientes expresiones:

$$\begin{aligned} X_a &= r_a (\cos \Omega_a \cos U_a - \sin \Omega_a \sin U_a \cos i_a), \\ Y_a &= r_a (\sin \Omega_a \cos U_a + \cos \Omega_a \sin U_a \cos i_a), \\ Z_a &= r_a \sin i_a \sin U_a. \end{aligned} \quad (5.8)$$

Actuando de manera similar con los elementos orbitales de la Tierra hallamos sus valores correspondientes, U_t , V_t y r_t , para cada grado en el mismo intervalo, y después las coordenadas rectangulares eclípticas heliocéntricas de la Tierra, X_t, Y_t, Z_t . Hecho todo lo anterior, es el momento de hallar la distancia Δ , que hay entre los dos cuerpos en el espacio, entre cada par de puntos tridimensionales, (X_a, Y_a, Z_a) y (X_t, Y_t, Z_t) , y para cada longitud eclíptica heliocéntrica $\lambda_a = \lambda_t$. Decir que, aunque yo calculé dichas coordenadas rectangulares para cada grado del intervalo $[0^\circ, 360^\circ]$, es decir, desde $\lambda_a = \lambda_t = 0^\circ$ hasta $\lambda_a = \lambda_t = 360^\circ$, si pensamos que Apophis tiene una pequeña inclinación de su órbita respecto a la eclíptica, es lógico suponer que el MOID se encuentre en el entorno de la longitud de su nodo ascendente $\Omega = 203.95667208^\circ$, por lo que puede servirnos trabajar sólo en el intervalo $[200^\circ, 210^\circ]$. Actuando así sobre este intervalo más estrecho, observamos que la mínima distancia Δ entre los dos cuerpos se obtiene cuando $\lambda_a = \lambda_t = 204^\circ$. A modo de ejemplo de aplicación, se van a desarrollar los cálculos anteriormente citados para los dos cuerpos y para una misma longitud orbital eclíptica heliocéntrica de $\lambda_a = \lambda_t = 204^\circ$.

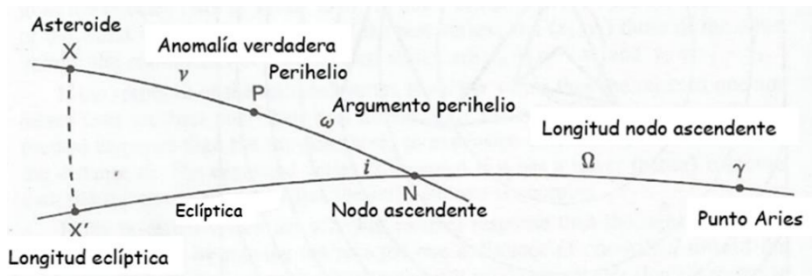


Figura 12. Elementos orbitales de un asteroide, en un momento dado.

5.1. Apophis

Argumento de latitud:

$$U_a = \lambda_a - \Omega_a = 204^\circ - 203.95667208^\circ = 0.04332792^\circ \quad (5.9)$$

Anomalía verdadera:

$$v_a = \lambda_a - \Omega_a - \omega_a = 204^\circ - 203.95667208^\circ - 126.60398809^\circ = -126.560660^\circ, \quad (5.10)$$

que, expresados en la rama $[0^\circ, 360^\circ]$, da $v_a = 233.4393398242^\circ$.

Radio vector Apophis–Sol:

$$r_a = \frac{a_a(1 - e_a^2)}{1 + e_a \cos V_a} = 1.00331929 \text{ UA}. \quad (5.11)$$

Coordenadas rectangulares eclípticas heliocéntricas:

$$\begin{aligned} X_a &= r_a(\cos \Omega_a \cos U_a - \sin \Omega_a \sin U_a \cos i_a) = -0.916578 \text{ UA}, \\ Y_a &= r_a(\sin \Omega_a \cos U_a + \cos \Omega_a \sin U_a \cos i_a) = -0.40808554 \text{ UA}, \\ Z_a &= r_a \sin i_a \sin U_a = +0.00004419 \text{ UA}. \end{aligned} \quad (5.12)$$

5.2. Tierra

Argumento de latitud:

$$U_t = \lambda_t - \Omega_t = 204^\circ - 174.81604597^\circ = 29.18395402^\circ \quad (5.13)$$

Anomalía verdadera:

$$v_t = \lambda_t - \Omega_t - \omega_t = 204^\circ - 174.81604597^\circ - 288.1977501^\circ = -259.013796^\circ, \quad (5.14)$$

que, expresados en la rama $[0^\circ, 360^\circ]$, da $v_t = 100.98620386^\circ$.

Radio vector Tierra–Sol:

$$r_t = \frac{a_t(1 - e_t^2)}{1 + e_t \cos V_t} = 1.00291383 \text{ UA}. \quad (5.15)$$

Coordenadas rectangulares eclípticas heliocéntricas:

$$\begin{aligned} X_t &= r_t (\cos \Omega_t \cos U_t - \sin \Omega_t \sin U_t \cos i_t) = -0.91620738 \text{ UA}, \\ Y_t &= r_t (\sin \Omega_t \cos U_t + \cos \Omega_t \sin U_t \cos i_t) = -0.40792181 \text{ UA}, \\ Z_t &= r_t \sin i_t \sin U_t = 0.00002640 \text{ UA}. \end{aligned} \quad (5.16)$$

Conocidas las coordenadas rectangulares de Apophis y de la Tierra, para una misma longitud eclíptica $\lambda_a = \lambda_t = 204^\circ$, y para una misma época y fecha, se puede hallar la distancia tridimensional en el espacio que los separa mediante la siguiente expresión:

$$\Delta = \sqrt{(X_a - X_t)^2 + (Y_a - Y_t)^2 + (Z_a - Z_t)^2} \quad (5.17)$$

Sustituyendo, $\Delta = 0.00040584 \text{ UA}$. Antes de aplicar el método geométrico simple, es aconsejable una serie de iteraciones en los valores de los pares $\lambda_a = \lambda_t$, con el objetivo de ir ajustando cada vez más los valores correctos del MOID.

Por ejemplo, podemos explorar qué valor adquiere Δ , para una $\lambda_a = 203.5^\circ$ y una $\lambda_t = 204^\circ$. Después de realizadas las operaciones necesarias vemos que $\Delta = 0.008969152 \text{ UA}$, mientras que para $\lambda_a = 204^\circ$ y $\lambda_t = 203.5^\circ$ el valor es $\Delta = 0.008769151 \text{ UA}$, mejor que el anterior par. Seguimos con $\lambda_a = 203.5^\circ$, $\lambda_t = 203.5^\circ$, dándonos una distancia $\Delta = 0.002124299 \text{ UA}$. Podemos continuar realizando más iteraciones en nuestra búsqueda de valores para Δ cada vez menores, pero en esencia ya estaríamos aplicando el método simple de cálculo, que se va a desarrollar de una forma más gráfica en el punto siguiente.

5.3. El método simple geométrico

En el apartado 2c de la Sección 4 ya se describió de forma general este método y su objetivo final de encontrar la menor distancia entre λ_a y λ_t , en UA. Ahora se van a presentar las normas a aplicar o los pasos concretos a dar para alcanzar ese MOID buscado. Asignamos como valores de los vértices del triángulo A, B, C, los dados anteriormente en las iteraciones. Así, el vértice $A = (X_1, Y_1) = (\lambda_a, \lambda_t) = (203.5^\circ, 204^\circ)$ es quien tiene el peor valor, pues $\Delta = 0.008969152 \text{ UA}$. El vértice $B = (X_2, Y_2) = (\lambda_a, \lambda_t) = (204^\circ, 204^\circ)$ tiene el mejor valor, para $\Delta = 0.00040584 \text{ UA}$. El vértice $C = (X_3, Y_3) = (\lambda_a, \lambda_t) = (203.5^\circ, 203.5^\circ)$ es el otro valor de la terna, con $\Delta = 0.002124299 \text{ UA}$. Decir que, en este triángulo inicial, los tres valores de λ_a no pueden ser iguales, ni tampoco los tres valores correspondientes de λ_t . Teniendo esta premisa en cuenta, sobre este triángulo inicial se pueden realizar sobre los vértices de pares (λ_a, λ_t) las siguientes acciones:

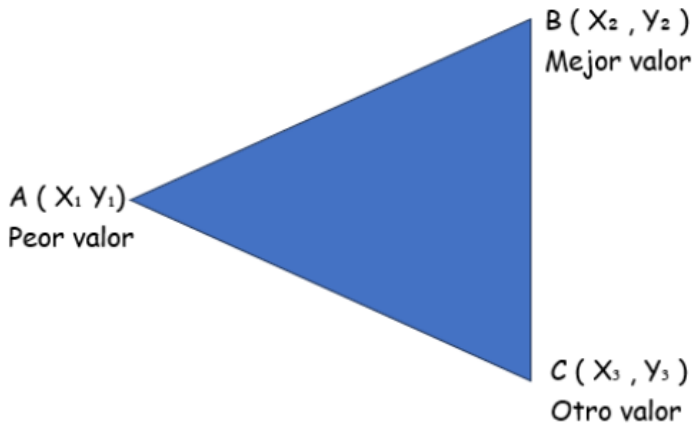


Figura 13. Triángulo método geométrico simple.

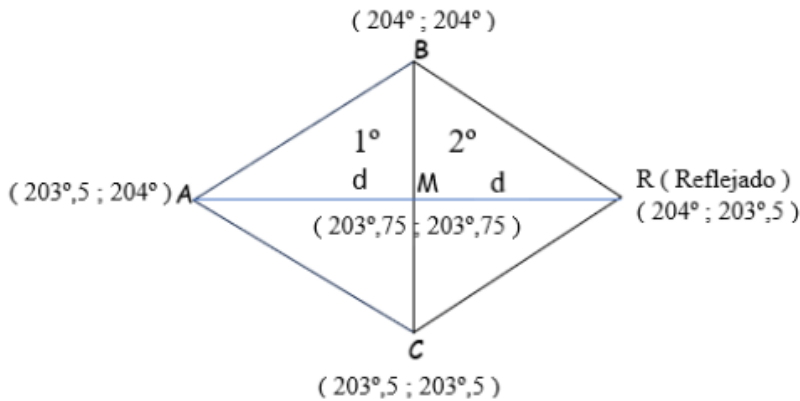


Figura 14. Vértice reflejado del A en el triángulo 1°.

- **Vértice reflejado (R).** Llamando d a la distancia AM, desde el punto A (el peor valor) hasta el punto M (punto medio del segmento BC), entonces el vértice reflejado de A se encuentra en el punto R, a una distancia d del punto medio M. Para hallar los valores del par de R (λ_a, λ_r) , se calcula así:

$$(X_2 + X_3 - X_1, Y_2 + Y_3 - Y_1) = (204^\circ + 203.5^\circ - 203.5^\circ, 204^\circ + 203.5^\circ - 204^\circ) = (204^\circ, 203.5^\circ).$$

Después, haciendo uso de las fórmulas 5.1–5.8, hallamos $U, V, |r|, X, Y, Z$, correspondientes tanto a Apophis como a la Tierra, y la distancia Δ entre los dos puntos de las órbitas de ambos cuerpos celestes, para esas respectivas longitudes eclípticas $(204^\circ, 203.5^\circ)$. Tenemos que Δ es igual a $\Delta = 0.008769151$ UA. Este nuevo vértice R se acepta si su Δ es mejor que el peor valor del anterior triángulo nº 1, como sucede en este caso. Si R fuera aún mejor que el mejor vértice (el B) del triángulo nº 1, se intenta una expansión $E = 2d$. Este vértice expandido se acepta si su Δ es mejor (es decir tiene un menor valor en UA) que el que tenía el punto A (el peor del triángulo nº 1). Si no se cumple este supuesto, entonces es rechazado. Ahora tenemos un nuevo triángulo, nº 2, con vértices R (el peor, $204^\circ, 203.5^\circ$) el B (el mejor, $204^\circ, 204^\circ$), y el C (el otro valor, $203.5^\circ, 203.5^\circ$).

Sucesivamente iríamos trabajando con los triángulos nº 2, 3, 4, etc., realizando sobre ellos las

acciones requeridas para lograr siempre la mejor terna de valores (A, B, C) cuyos valores de Δ sean los mejores, es decir que tengan las distancias Δ cada vez menores en UA.

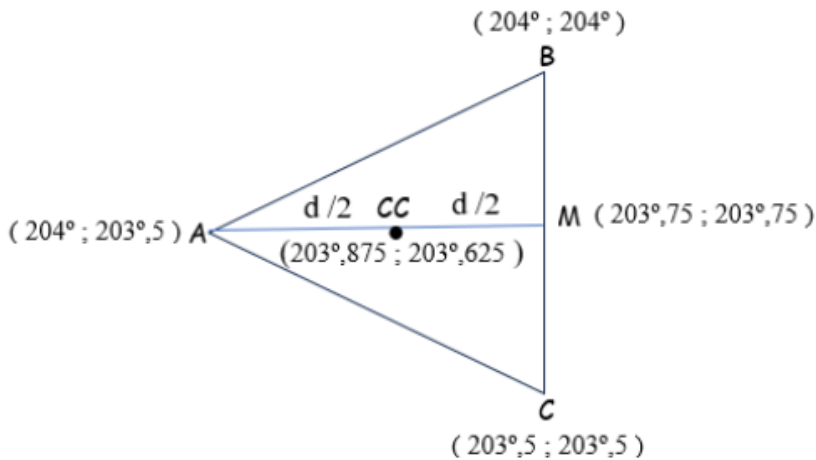


Figura 15. Vértice contraído del A en el triángulo nº 2.

- **Vértice contraído (CC).** Si el vértice reflejado R del par A en el triángulo nº 1 hubiera tenido una distancia Δ , en UA peor que el valor de A, entonces se tiene que realizar una contracción (CC), llevando este vértice A hacia M (punto medio entre los vértices B y C), una distancia $d/2$, para obtener un nuevo vértice para el nuevo triángulo. Este punto CC es aceptado si tiene una Δ mejor que el A (el peor del triángulo anterior, nº 2). Si esto no se produce, y no es mejor que el vértice A, entonces se realiza una reducción (R').
- **Vértices reducidos (R').** Si se dan las condiciones anteriores, esta *reducción* se lleva a cabo acercando el vértice A (el de peor valor) hacia el B (el mejor valor) hasta alcanzar su punto medio, es decir, una distancia $AB/2$. De igual modo se hace lo mismo acercando el vértice C hacia el B una distancia igual a $CB/2$, alcanzando el punto M (ya hallado antes). Si uno –o los dos– vértices reducidos toma o toman un mejor valor en la distancia Δ , entonces es/son aceptado/s.

En el triángulo nº 15 de la Fig. 16 se puede ver una reducción llevada a cabo sobre el vértice A hacia el B, que fue aceptada, mientras que esa reducción sobre el vértice C no fue aceptada. Antes se habían realizado sobre A una reflexión (R) y una contracción (CC), que no fueron aceptados por ofrecer un valor de Δ mayor (en UA) que el que tenía A. De esta forma el triángulo nº 16 tiene como vértices nuevos, C (el peor), B (el mejor) y R' (otro valor). Estas actuaciones sobre cada uno de los vértices en los sucesivos triángulos de pares (λ_a, λ_t) están recogidos gráficamente en las Figs. 14, 15 y 16, para una mejor comprensión de las mismas.

En el Material Suplementario 2 se presentan de una forma detallada y concreta todos los triángulos que han sido necesarios para obtener el MOID entre las órbitas de Apophis y la Tierra, especificando la acción de aproximación realizada sobre cada vértice y sobre cada triángulo. Aunque el número de triángulos (veintitrés) puedan resultar excesivos, para ser presentados todos en el Material Suplementario 2, creo necesario que figuren en su totalidad, ya que pueden servir como guía práctica en el cálculo del MOID entre dos cuerpos celestes, con órbitas elípticas, que comparten el mismo foco, el Sol, ya sean asteroides, cometas o planetas del Sistema Solar, cumpliendo también de esta forma con un objetivo

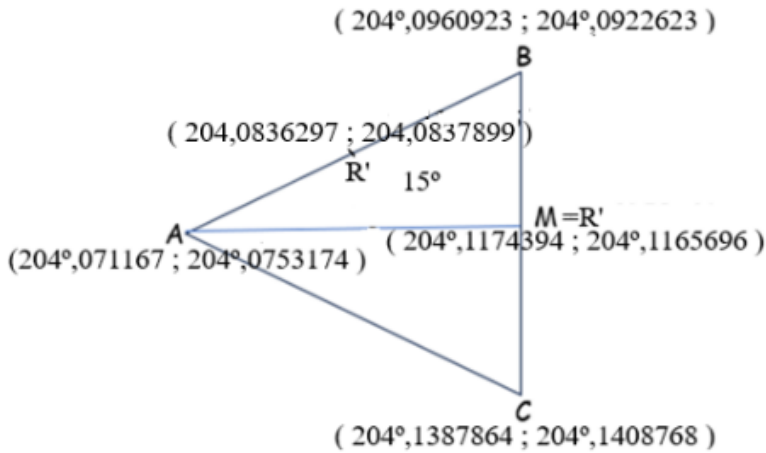


Figura 16. *Vértices reducidos de los vértices A y C hacia el vértice B, en el triángulo nº 15.*

general en todos mis trabajos astronómicos : presentar de la forma más didáctica posible los contenidos a desarrollar, pues algunos pueden resultar un poco complicados de entender y aplicar, por parte del lector, en una cuestión de alguna investigación que implique el uso de estos mismos conceptos.

Analizando los resultados obtenidos en el MOID, vemos que la alarma desatada respecto a la posible colisión de Apophis con la Tierra, tomando como elementos orbitales los que figuraban para el 13 de septiembre de 2023, estaba más que justificada. La órbita del asteroide se aproxima a la de la Tierra en una distancia $\Delta = 0.000136116$ UA, es decir, a una distancia del centro de nuestro planeta $d = 20362.6$ km, correspondiente a unos 3.2 radios ecuatoriales terrestres, por lo que se halla a unos 14000 km sobre la superficie que pisamos en nuestra vida diaria. Menos mal que las nuevas mediciones realizadas sobre la órbita de Apophis, parecen descartar la probabilidad de que ocurra esta catástrofe, ...¡por el momento!

6. Distancia a la Tierra y Coordenadas de Apophis en la fecha de estudio

Dicho lo anterior, ahora entendemos el por qué de catalogar estos cuerpos celestes como potencialmente peligrosos para la Humanidad y asumir la necesidad de idear una serie de misiones espaciales que nos hagan sentir más seguros respecto a ellos. En este apartado se van a desarrollar los cálculos matemáticos realizados para evaluar la distancia, r_a , a la que pasará Apophis el día 13 de abril de 2029, a las 21h 44m UT, así como por dónde va a ser posible verlo a simple vista, por parte de un observador situado en el Real Observatorio de Madrid, situado en pleno centro de la ciudad de la capital española. Nuevamente aclarar la diferencia entre el MOID y la distancia Apophis-Tierra en un momento dado. Aquel nos informa de cuánto se acercan sus órbitas en el espacio mientras que la distancia r_a mide cuánto están separados los dos cuerpos en un momento concreto de tiempo.

6.1. Cálculo de la distancia Tierra–Apophis

Para hallar la distancia entre los dos cuerpos se tienen que dar los siguientes pasos de cálculo:

- Hallar las coordenadas eclípticas heliocéntricas de la Tierra, la longitud L y la latitud B , sus coordenadas cartesianas, también heliocéntricas (X_t, Y_t, Z_t) , junto con el módulo del radio vector Tierra–Sol, $|R_t|$, todo ello para la fecha de máximo acercamiento y para el equinoccio medio de la misma.

- b. Obtener los elementos orbitales de Apophis (J2000.0), para la fecha de su máxima aproximación (*osculating elements*), directamente del banco de datos del JPL.
- c. Calcular las coordenadas eclípticas heliocéntricas de Apophis, longitud λ_a , y latitud β_a , sus coordenadas cartesianas, también heliocéntricas (X_a, Y_a, Z_a) , junto con el módulo del radio vector Apophis–Sol $|R_a|$, todo en el sistema de referencia J2000.0, y para la fecha de máximo acercamiento y el equinoccio medio de la misma.
- d. Reducir los elementos orbitales del asteroide que varían al pasar desde el sistema de referencia J2000.0 al correspondiente a la fecha de estudio. Estos elementos que varían son la inclinación i_a , la longitud del nodo ascendente Ω_a y el argumento del perihelio ω_a . El semieje mayor a_a y la excentricidad e_a no se ven afectados, ya que son casi iguales en el año 2000 que en el 2029, pues el tamaño y forma de la órbita no cambia.
- e. Hallar la distancia Apophis–Tierra en el momento de máxima proximidad a su superficie.

A continuación se desarrollan más en detalle cada uno de los pasos anteriores correspondientes a la Tierra y a Apophis.

Para la Tierra:

Coordenadas eclípticas heliocéntricas L y B . Para hallar estas coordenadas nos servimos de las series de términos periódicos para la Tierra, ofrecidas por Bretagnon y Francou, del ‘Bureau des Longitudes’ de París, en su teoría VSOP87 (*Variations Seculaires des Orbites Planetaries*). Las sumas de estos términos son evaluados para un momento concreto de tiempo y para el equinoccio medio de esa fecha. Existen series para cada uno de los planetas del Sistema Solar (ver Material Suplementario 3).

Las series L_0, L_1, L_2, \dots , sirven para calcular la longitud eclíptica heliocéntrica (no la longitud orbital del planeta que se mide en dos planos, el de la eclíptica y el propio de la órbita del planeta). Las series $\beta_0, \beta_1, \beta_2, \dots$, sirven para la latitud eclíptica, y las R_0, R_1, R_2, \dots , para el radio vector R . Cada línea horizontal en las series representa a un término periódico, conteniendo cuatro números, más las dos columnas, correspondientes a la serie y el planeta del que se trata. El primer número nos informa del número de orden de cada término, para cada una de las series, mientras los tres restantes son las cantidades A, B y C , correspondientes a cada término. El valor final que adquiere cada término viene dado en función de estos números, distintos para cada uno de ellos y, también en función del tiempo T transcurrido desde la época J2000.0 (1 de enero de 2000 a las 12h), expresado en milenios. Para el 13 de abril de 2029, a las 21h 44m 13s UT = 21h 46m TDB = 2462240.40571 JD, el valor que adquiere T es igual a $T = (2462240.40571 - 2451545.0)/365250 = 0.029282424942$ milenios. El valor adquirido en cada término está dado por $A \cos(B + CT)$, en el que B y C están expresados en radianes y el coeficiente A está en unidades de 10^{-8} radianes (para la longitud L y la latitud B), mientras que el radio vector R lo hace en unidades de 10^{-8} unidades astronómicas (UA). A modo de ejemplo de aplicación, el término nº 10 de la serie L_0 es igual a:

$$\begin{aligned} A \cos(B + CT) &= 1273 \cos(2.0371 + 529.691 \times 0.029282424942) \\ &= 338.2942787/10^8 \text{ radianes.} \end{aligned} \quad (6.1)$$

El cálculo de la longitud eclíptica heliocéntrica de la Tierra, para la fecha de estudio y el equinoccio medio de la misma, se logra del modo siguiente:

$$L = (L_0 + L_1T + L_2T^2 + L_3T^3 + L_4T^4 + L_5T^5)/10^8. \quad (6.2)$$

En esta expresión, L_0 es el sumatorio de todos los términos de la serie L_0 , L_1 es el sumatorio de los términos de la serie L_1 , etc. El resultado final para L , sustituyendo valores, es igual a:

$$L = 185.7771023401 \text{ radianes} = 10644.24389^\circ \quad (6.3)$$

que, expresados en el intervalo $[0^\circ, 360^\circ]$, da $10644.24389^\circ/360^\circ = 29.5673441507$ vueltas de 360° , que equivalen a $L = 204.2438942584^\circ$.

La latitud eclíptica heliocéntrica es igual a

$$B = (B_0 + B_1T)/10^8 = -0.0001721773^\circ. \quad (6.4)$$

El radio vector es

$$R_t = (R_0 + R_1T + R_2T^2 + R_3T^3 + R_4T^4)/10^8 = 1.0029225514 \text{ UA}. \quad (6.5)$$

Coordenadas cartesianas eclípticas heliocéntricas (X_t, Y_t, Z_t). Conocidas la longitud y la latitud eclípticas heliocéntricas, además de la distancia Tierra– Sol, se pueden hallar sus coordenadas cartesianas eclípticas heliocéntricas (X_t, Y_t, Z_t), haciendo uso de las siguientes expresiones:

$$\begin{aligned} X_t &= R_t \cos B \cos L = -0.914470606 \text{ UA}, \\ Y_t &= R_t \cos B \sin L = -0.411821751 \text{ UA}, \\ Z_t &= R_t \sin B = -0.000003013 \text{ UA}. \end{aligned} \quad (6.6)$$

Para Apophis:

Aunque para hallar el MOID se trabajó con los elementos orbitales de Apophis para la época del 13 de septiembre de 2023, a las 0h TDB, para hallar la distancia a la que pasará de nosotros el día de su máximo acercamiento, el 13 de abril de 2029, a las 21h 44m 13s UT, es necesario tener estos elementos orbitales correspondientes a ese momento en concreto, ya que el campo gravitatorio de la Tierra está haciendo variar continuamente alguno de los mismos, en función de la distancia del asteroide a la misma, alterando su trayectoria. En la Tabla de la Fig. 6 ya se recogieron los elementos orbitales correspondientes al 13 de abril de los años 2028, 2029 y 2036 (este último otro año catalogado como crítico, por presentar cierto riesgo de impacto). Estos elementos orbitales de Apophis durante su trayectoria de 2029 se mantienen con unos valores similares hasta un día antes, el 12 de abril, observándose después un cambio significativo, con aumento en la excentricidad e , semieje mayor a , argumento de latitud del perihelio ω , inclinación de la órbita i y del periodo P ; permaneciendo casi invariable la longitud del nodo ascendente Ω . Estas variaciones, algunas muy intensas, se observan claramente en el momento del máximo acercamiento a la Tierra, el 13 de abril a las 21h 44m 13s UT = 21h 46m TDB. Con el objetivo de visualizar los cambios que tendrán lugar en el asteroide a medida que se va aproximando a la órbita de la Tierra, a continuación se van a dar los elementos correspondientes al 13 de marzo y al 13 de abril del 2029, es decir con un intervalo de un mes.

Para poder entender bien la leyenda de los mismos correctamente es aconsejable cotejar los datos entre la Tabla de la Fig. 18 y la Tabla de la Fig. 6, con la intención de interpretar bien los datos en la fecha del máximo acercamiento de abril del 2029. En la Fig. 19 se puede ver un gráfico representando los principales elementos orbitales de un cuerpo celeste que mantiene una órbita elíptica alrededor del Sol. En la Tabla de la Fig. 18 y en la Fig. 19 los elementos orbitales que aparecen en cada cuadro son:

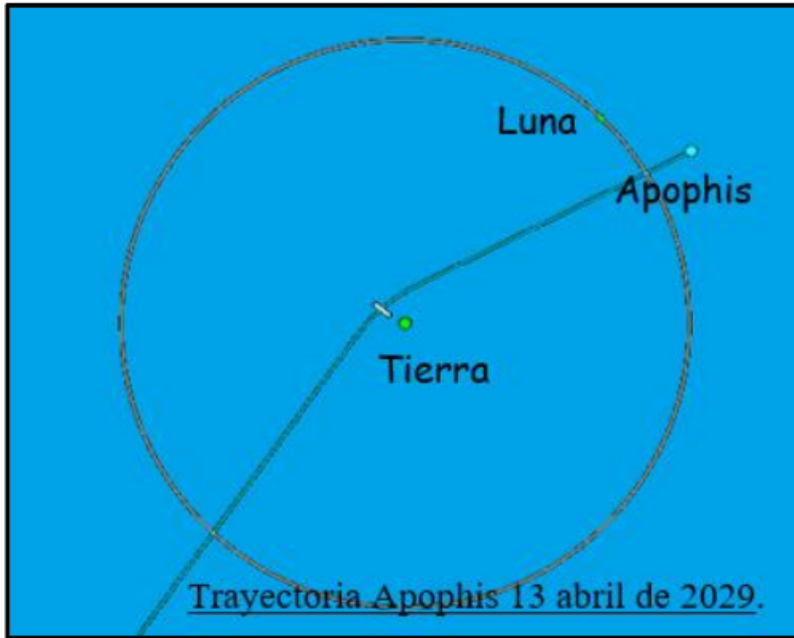


Figura 17. Variación de la órbita de Apophis en el momento de su máximo acercamiento a la Tierra.

DJ...2462209.406944444 = A.D. 2029-Mar-13 21:46:00.0000 TDB
 EC=1.912524553427472E-01 QR= 7.459140576001601E-01..... IN=3.342486774398328E+00
 OM= 2.038586354253304E+02... W = 1.267021998196493E+02..... Tp= 2462337.122014564928
 N = 1.112731413150203E+00.....MA= 2.178874295440699E+02.....TA= 2.066451070678725E+02
 A = 9.223076626665722E-01.....AD= 1.098701267732984E+00.....PR= 3.235282079264936E+02

DJ...2462240.406944444 = A.D. 2029-Apr-13 21:46:00.0000 TDB
 EC= 2.236577634208273E-01.....QR= 7.905122923951475E-01.....IN= 3.625637694579639E+00
 OM= 2.037588105709435E+02.....W = 9.920415910981717E+01.....Tp= 2462316.887706857640
 N = 9.592260271647556E-01.....MA= 2.866376621158566E+02.....TA= 2.608589252870700E+02
 A = 1.018252331443942E+00.....AD= 1.245992370492737E+00.....PR= 3.753025770829786E+02

Figura 18. Elementos orbitales de Apophis en marzo y abril de 2029.

EC=Excentricidad de la órbita	QR= Distancia del perihelio	IN= Inclinación de la órbita
OM=Longitud nodo ascendente	W = Argumento del perihelio	Tp= Fecha último paso perihelio
N = Movimiento medio diario	MA=Anomalía media	TA= Anomalía verdadera
A = Semieje mayor de la órbita	AD= Distancia del afelio	PR= Periodo orbital del cuerpo

Para hallar las coordenadas eclípticas heliocéntricas de Apophis, (λ_a, λ_t) , primero tenemos que conocer sus coordenadas cartesianas eclípticas heliocéntricas (X_a, Y_a, Z_a) , en el sistema de referencia J2000.0, como para la Tierra, y su radio vector (R_a) , o distancia al Sol en el momento de estudio. Antes

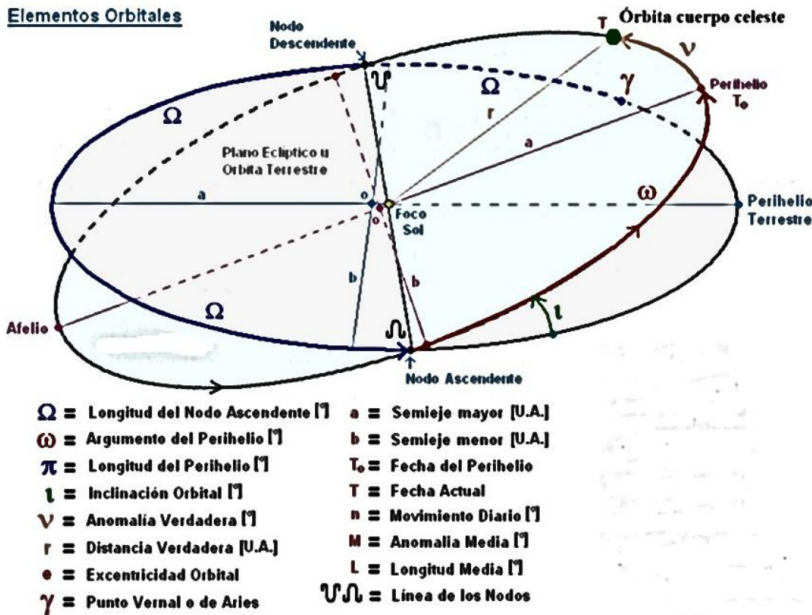


Figura 19. Elementos orbitales de un cuerpo celeste.

de abordar el cálculo de estas dos coordenadas heliocéntricas (cartesianas y eclípticas), primero es necesario conocer el argumento de latitud de asteroide (U_a), que corresponde a la suma del argumento del perihelio ω_a más la anomalía verdadera V_a , ambos conocidos. Tenemos:

$$U_a = \omega_a + V_a = 99.2041738541^\circ + 260.8589107495^\circ = 360.0630846036^\circ \quad (6.7)$$

que, expresado en el intervalo $[0^\circ, 360^\circ]$, nos queda $U_a = 0.0630846036^\circ$.

A continuación se halla la distancia Apophis–Sol (R_a), en función del semieje mayor (a_a), la excentricidad (e_a) y la anomalía verdadera (V_a) del asteroide:

$$R_a = \frac{a_a(1 - e_a^2)}{1 + e_a \cos V_a} = \frac{1.0182522627(1 - 0.2236577328^2)}{1 + 0.2236577328 \cos 260.8589107495^\circ} = 1.0029530193 \text{ UA.} \quad (6.8)$$

Ahora ya estamos en condiciones de calcular sus coordenadas cartesianas eclípticas heliocéntricas (X_a, Y_a, Z_a) mediante las siguientes expresiones :

$$\begin{aligned} X_a &= R_a (\cos \Omega_a \cos U_a - \sin \Omega_a \sin U_a \cos i_a) = -0.9175077191 \text{ UA,} \\ Y_a &= R_a (\sin \Omega_a \cos U_a + \cos \Omega_a \sin U_a \cos i_a) = -0.4050855953 \text{ UA,} \\ Z_a &= R_a \sin i_a \sin U_a = 0.0000698318 \text{ UA} \end{aligned} \quad (6.9)$$

Conocidas estas coordenadas cartesianas (X_a, Y_a, Z_a) se pueden hallar las coordenadas eclípticas heliocéntricas de Apophis (λ_a, β_a).

Para la Longitud Eclíptica λ_a :

$$\tan \lambda_a = \frac{Y_a}{X_a} = \frac{-0.40508559}{-0.91750771} = 0.41576830. \quad (6.10)$$

Teniendo en cuenta que λ_a se encuentra en el tercer cuadrante $[180^\circ, 270^\circ]$, pues $Y_a < 0$ y $X_a < 0$, obtenemos $\lambda_a = 203.8217686952^\circ$.

Para la Latitud Eclíptica β_a :

$$\sin \beta_a = \frac{Z_a}{R_a} = \frac{0.0000698318}{1.002953019} = 0.00006962619, \quad (6.11)$$

y tomando el arcoseno obtenemos $\beta_a = 0.0039892861^\circ$. Este pequeño valor es lógico, pues sabemos que β_a se halla en el intervalo $[0^\circ, 180^\circ]$, no muy separado del plano de la eclíptica.

Tanto las coordenadas cartesianas (X_a, Y_a, Z_a) , como la longitud eclíptica (λ_a) y la latitud eclíptica (β_a) están expresadas respecto al marco de referencia J2000.0. Llegados aquí, ya se han resuelto los pasos a, b, y c del apartado 6.1 (cálculo de la distancia Δ Tierra–Apophis) de la Sección 6.

Ya que las coordenadas cartesianas de la Tierra (X_t, Y_t, Z_t) en la fecha del máximo acercamiento, están referidas respecto al equinoccio medio en ese momento y las de Apophis (X_a, Y_a, Z_a) lo están respecto al marco de referencia J2000.0, se hace necesario reducir los elementos orbitales del asteroide $(i_0, \omega_0, \Omega_0)$, pues varían al pasar desde el sistema J2000.0 al correspondiente a la fecha de estudio (13 de abril de 2029 a las 21h 46m TDB = 21h 44m 13s UT), paso d del punto 6.1, Sección 6.

Los pasos a realizar para hallar la inclinación (i'), el argumento del perihelio (ω') y la longitud del nodo ascendente (Ω') correspondientes a este instante temporal final son:

1. Calcular el tiempo transcurrido entre la época J2000.0 y la fecha final de estudio, expresado en siglos julianos:

$$t = \frac{(JD - JD_{J2000.0})}{36525} = \frac{2462240.40571 - 2451545.0}{36525} = 0.29282424942 \text{ siglos.} \quad (6.12)$$

2. Calcular los ángulos precesionales η , Π y p , siendo η el ángulo entre la eclíptica, en la época inicial, y la eclíptica final; y Π y p los otros dos ángulos precesionales eclípticos:

$$\eta = (47.0029'' - 0.06603''T + 0.000598''T^2)t + 0.00006''t^3(-0.03302'' + 0.000598''T)t^2, \quad (6.13)$$

en donde $T = (JD_{\text{inicial}} - JD_{\text{final}})/36525$, es decir, siglos transcurridos desde la época inicial y la final. Si la época inicial es J2000.0, como ocurre en este caso, entonces $T = 0$ y $\eta = 13.76075909''$ que, dividiendo entre $3600''/^\circ$, da $\eta = 0.00382243^\circ$.

$$\Pi = 174.876384^\circ + 3289.4789''T + 0.60622''T^2 - (869.8089'' + 0.50491''T)t + 0.03536''t^2, \quad (6.14)$$

pero si la época inicial es J2000.0, entonces

$$\Pi = 174.876384^\circ - 869.8089''t + 0.03536''t^2 = 174.8056345^\circ. \quad (6.15)$$

Hay que tener cuidado porque el segundo y tercer sumandos están en segundos de arco.

$$p = (5029.0966'' + 2.22226''T - 0.000042''T^2)t + (1.11113'' - 0.000042''T)t^2 - 0.000006''t^3. \quad (6.16)$$

Si la época inicial es J2000.0, entonces

$$p = 5029.0966''t + 1.11113''t^2 - 0.000006''t^3 = 0.40909353^\circ \quad (6.17)$$

3. Hallar las (A', B', C') en función de los anteriores ángulos precesionales (η , Π , p) y de la longitud eclíptica heliocéntrica λ_a y la latitud eclíptica heliocéntrica β_a , halladas con anterioridad

($\lambda_a = 203.8217686952^\circ$ y $\beta_a = 0.0039892861^\circ$):

$$\begin{aligned} A' &= \cos \eta \cos \beta_a \sin (\Pi - \lambda_a) - \sin \eta \sin \beta_a = -0.485055891, \\ B' &= \cos \beta_a \cos (\Pi - \lambda_a) = 0.874483151, \\ C' &= \cos \eta \sin \beta_a + \sin \eta \cos \beta_a \sin (\Pi - \lambda_a) = 0.0000372661. \end{aligned} \quad (6.18)$$

Para hallar la nuevas longitud, λ'_a , y latitud β'_a eclípticas heliocéntricas correspondientes a la fecha de estudio, durante el máximo acercamiento, se usan las siguientes expresiones:

$$\begin{aligned} \tan (p + \Pi - \lambda'_a) &= A'/B' = \frac{-0.485055891}{0.874483151} = -0.55467723 \\ \rightarrow p + \Pi - \lambda'_a &= -29.01613435^\circ. \end{aligned} \quad (6.19)$$

Despejando, $\lambda'_a = \Pi + 29.01613435^\circ = 204.2308624^\circ$. Además:

$$\sin \beta'_a = C' = 0.0000372661 \rightarrow \beta'_a = 0.0021351924^\circ. \quad (6.20)$$

Ya tenemos halladas las coordenadas eclípticas heliocéntricas de Apophis el 13 de abril de 2029 a las 21h 44m 13s UT.

4. Calcular los elementos i' (inclinación), ω' (argumento del perihelio) y Ω' (longitud del nodo ascendente), reducidos al equinoccio medio de la fecha de paso del asteroide en su máximo acercamiento a la Tierra. Para ello se hace uso de las siguientes expresiones:

Inclinación i' :

$$\cos i' = \cos i \cos \eta + \sin i \sin \eta \cos (\Omega - \Pi), \quad (6.21)$$

en donde $i = 3.625637694^\circ$ y $\Omega = 203.7588103533^\circ$, los valores que tenían en el sistema de la época J2000.0, y η y Π son los ángulos precesionales hallados antes. Sustituyendo valores obtenemos $\cos i' = 0.998002221$, y de aquí $i' = 3.6222934779^\circ$.

Longitud del nodo ascendente Ω' :

Haciendo uso del nuevo ángulo auxiliar $\Psi = \Pi + p = 174.8056345^\circ + 0.40909353^\circ = 175.21472806^\circ$ y de la inclinación i' , hallada antes, tenemos que

$$\sin i' \sin (\Omega' - \Psi) = \sin i \sin (\Omega - \Pi) = 0.030612741, \quad (6.22)$$

de donde $\sin (\Omega' - \Psi) = 0.484541023$ y $\Omega' = 204.1971339152^\circ$.

Argumento de latitud del perihelio ω' :

El nuevo argumento de perihelio (ω') es igual al argumento de perihelio J2000.0 más una corrección $\Delta\omega$. Para hallarla nos ayudamos de las siguientes expresiones matemáticas:

$$\sin i' \sin \Delta\omega = -\sin \eta \sin (\Omega - \Pi) = -0.0000322959. \quad (6.23)$$

Despejando, $\sin \Delta\omega = -0.000511182$, y su arcoseno es $\Delta\omega = -0.0292888597^\circ$. Ya que $\sin \Delta\omega < 0$, el ángulo puede estar en el tercer, $[180^\circ, 270^\circ]$ o cuarto, $[270^\circ, 360^\circ]$, cuadrantes. Para resolver esta incertidumbre nos ayudamos de la expresión:

$$\sin i' \cos \Delta\omega = \sin i \cos \eta - \cos i \sin \eta \cos (\Omega - \Pi) = 0.063178833. \quad (6.24)$$

Despejando, $\cos \Delta\omega = 0.999999869$, con arcoseno $\Delta\omega = 0.0292888597^\circ$. Como $\cos \Delta\omega > 0$, hemos descubierto que $\Delta\omega$ está en el cuarto cuadrante, y expresando su valor en grados positivos, $\Delta\omega = 359.9707114^\circ$. Entonces $\omega' = \omega + \Delta\omega = 99.1748852^\circ$. Los nuevos valores de los elementos orbitales de Apophis, respecto al equinoccio medio de la fecha de estudio, son:

$$\begin{aligned} \text{Inclinación, } i' &= 3.622293477^\circ \\ \text{Longitud del nodo ascendente, } \Omega' &= 204.1971339152^\circ \\ \text{Argumento del perihelio, } \omega' &= 99.1748852^\circ. \end{aligned} \tag{6.25}$$

Estos nuevos valores nos permitirán ajustar más las coordenadas cartesianas heliocéntricas (X_a, Y_a, Z_a) de Apophis y, por consiguiente, nos permitirán hallar con más exactitud la distancia Tierra–Apophis en su momento de máximo acercamiento a la superficie de nuestro planeta.

7. Cálculo revisado de la distancia real Apophis–Tierra

Para realizar este cálculo, de nuevo tenemos que repetir los pasos dados en el punto c del apartado 6.1 de la Sección 6, para hallar las coordenadas (X_a, Y_a, Z_a) , utilizando los nuevos elementos orbitales i', Ω', ω' . Estas modificaciones se justifican porque, tanto las coordenadas de la Tierra como las de Apophis tienen que estar referenciadas respecto al mismo marco tridimensional (X, Y, Z) en el espacio y este debe ser el de 2029.

Los elementos orbitales del asteroide con los que se va a calcular la distancia Apophis- Tierra incorporan los valores i', Ω' y ω' , más aquellos que no han sufrido ninguna variación con el cambio del marco de referencia desde el equinoccio medio J2000 al equinoccio medio de la fecha de estudio en su máximo acercamiento. Tanto la distancia al Sol R_a como el semieje mayor a , la excentricidad e y la anomalía verdadera V_a conservan los mismos valores que en la época J2000.0. En la Tabla de la Fig. 20 se ven los valores de los elementos orbitales con los que se van a realizar los cálculos.

Elemento orbital de Apophis el 13-IV-2029	Valor
Inclinación de la órbita (i'_a)	3°,622293477
Longitud del nodo ascendente (Ω'_a)	204°,1971339152
Argumento del perihelio (ω'_a)	99°,1748852
Semieje mayor (a_a)	1,0182522627
Excentricidad (e_a)	0,2236577328
Anomalía verdadera (V_a)	260°,8589107495
Distancia al Sol (R_a)	1,0029530193 U.A.

Figura 20. Elementos orbitales de Apophis el 13 de abril de 2029, a las 21h 44m 13s UT.

Lo primero es conocer el argumento de latitud de asteroide (U_a), que corresponde a la suma del argumento del perihelio (ω'_a) más la anomalía verdadera (V_a):

$$U_a = \omega'_a + V_a = 99.1748852^\circ + 260.8589107495^\circ = 360.0337959^\circ \tag{7.1}$$

que, expresados en el intervalo $[0^\circ, 360^\circ]$, nos da $U_a = 0.337960066^\circ$. A continuación hallamos las coordenadas cartesianas eclípticas heliocéntricas:

$$\begin{aligned} X_a &= R_a (\cos \Omega'_a \cos U_a - \sin \Omega'_a \sin U_a \cos i'_a), \\ Y_a &= R_a (\sin \Omega'_a \cos U_a + \cos \Omega'_a \sin U_a \cos i'_a), \\ Z_a &= R_a \sin i'_a \sin U_a. \end{aligned} \quad (7.2)$$

Sustituyendo valores,

$$\begin{aligned} X_a &= -0.9145920338 \text{ UA}, \\ Y_a &= -0.4116262494 \text{ UA}, \\ Z_a &= +0.0000373762 \text{ UA}. \end{aligned} \quad (7.3)$$

Teniendo en cuenta que las coordenadas cartesianas eclípticas heliocéntricas de la Tierra son

$$\begin{aligned} X_t &= -0.9144706061 \text{ UA}, \\ Y_t &= -0.4118217511 \text{ UA}, \\ Z_t &= -0.0000030138 \text{ UA}. \end{aligned} \quad (7.4)$$

Conocidas las coordenadas cartesianas de los dos cuerpos en el espacio, la distancia Δ entre ellos se halla mediante la expresión (5.17):

$$\Delta = \sqrt{(X_a - X_t)^2 + (Y_a - Y_t)^2 + (Z_a - Z_t)^2} \quad (7.5)$$

Sustituyendo valores, tenemos que la distancia entre ambos es $\Delta = 0.00023366 \text{ UA}$. Si multiplicamos esta cantidad por el valor de 1 UA (149597870 km) nos da que la distancia de Apophis al centro de la Tierra es $\Delta = 34955 \text{ km}$. Esto significa que, cuando cruce, en ese momento, a gran velocidad ($\sim 7 \text{ km/s}$) el cielo nocturno, y a unos 29000 km sobre nuestras cabezas, justo un poco por debajo de las órbitas que mantienen nuestros satélites geostacionarios, entonces Apophis adquirirá una magnitud de 3.1, siendo visible a simple vista, alcanzando entonces una velocidad angular de unos $42^\circ/\text{h}$, atravesando la proyección de su trayectoria amplias zonas del Océano Índico, África, Océano Atlántico hasta finalmente alcanzar los Estados Unidos, siendo visible también desde Europa Y Asia (Fig. 21). Para valorar de una forma gráfica esta distancia, hemos de tener en cuenta que Apophis pasará entre la Tierra y la Luna casi a 1/10 de la distancia media entre ambos astros.

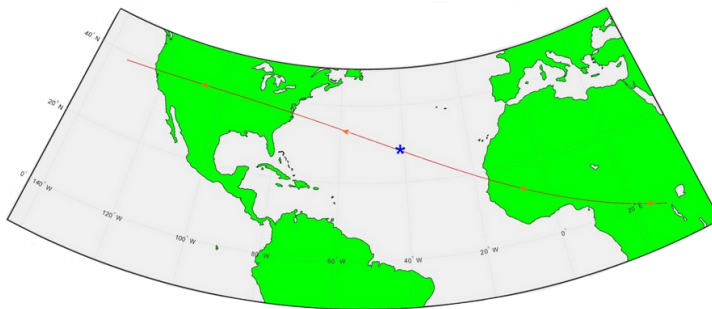


Figura 21. Proyección de la trayectoria de Apophis sobre la superficie terrestre en 2029 [6].

8. Localización de Apophis en el cielo

Hasta ahora se ha estudiado el asteroide fundamentalmente en su aspecto dinámico, realizando los cálculos necesarios para saber cuánto y dónde su órbita se acerca a la Tierra, es decir, hallando su MOID. También se ha calculado cuánto de cerca se hallará de nosotros en abril de 2029, pero para un observador instalado en la superficie de la Tierra, por ejemplo en la ciudad de Madrid, surge la pregunta: ¿en qué lugar del cielo nocturno madrileño será visible Apophis cuando se encuentre a su distancia mínima de nosotros esa tarde-noche? Ya que el objetivo es localizar en el cielo su posición, entre las constelaciones de estrellas, hay que hallar sus coordenadas celestes respecto al equinoccio medio de la fecha (2029).

Como estas coordenadas no han variado mucho respecto al sistema de referencia J2000.0, ya que el cambio por precesión no es muy significativo en ese intervalo de tiempo, no hay ningún problema en situarlo sobre una carta estelar o planisferio celeste para el equinoccio medio del año 2000. De este modo, los lectores pueden buscar su posición en cualquier planisferio convencional en uso hoy en día, como el que existe para la latitud 40° norte. En la Tabla de la Fig. 22 se recogen las coordenadas eclípticas heliocéntricas de los dos cuerpos celestes, respecto al equinoccio medio del año 2000 y las correspondientes al año 2029, en el momento de máximo acercamiento del asteroide. La distancia al Sol de la Tierra y de Apophis no ha variado porque su órbita era la misma.

Cuerpo celeste	Equinoccio medio año 2000		Equinoccio medio año 2029		Distancia al Sol U.A.
	Longitud eclíptica	Latitud eclíptica	Longitud eclíptica	Latitud eclíptica	
Apophis	203,82176869	0,0039892861	204,23085813	0,002134922	1,0029530193
Tierra	203,8348006	0,0016829195	204,2450999	-0,00017216	1,0029225514

Figura 22. Coordenadas eclípticas heliocéntricas de Apophis y el Sol en el año 2000 y en 2029.

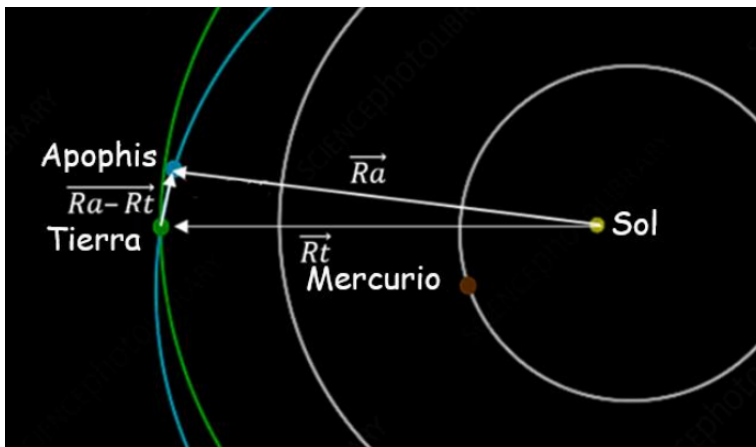


Figura 23. Transformación de coordenadas heliocéntricas a geocéntricas (resta de vectores).

Las coordenadas que se van a hallar serán las siguientes:

Coordenadas eclípticas geocéntricas (λ_a, β_a).

Teniendo las coordenadas heliocéntricas, tanto de Apophis como de la Tierra, para una misma fecha y en el mismo plano de referencia (la eclíptica), además de su radio vector o distancia al Sol, podemos hacer una transformación de coordenadas desde el punto de vista heliocéntrico al geocéntrico, que es como nosotros lo vemos. El mecanismo es realizar una resta de vectores $R_a - R_t$ para hallar las componentes cartesianas (X_a, Y_a, Z_a) , usando las siguientes expresiones:

$$\begin{aligned} X_a &= R_a \cos \beta_a \cos \lambda_a - R_t \cos \beta_t \cos \lambda_t = -0.0001301244 \text{ UA}, \\ Y_a &= R_a \cos \beta_a \sin \lambda_a - R_t \cos \beta_t \sin \lambda_t = +0.0002148125 \text{ UA}, \\ Z_a &= R_a \sin \beta_a - R_t \sin \beta_t = +0.000040385 \text{ UA}, \end{aligned} \quad (8.1)$$

donde se han sustituido los valores de la Tabla de la Fig. 22 correspondientes al equinoccio medio 2029.

Conocidas las componentes (X_a, Y_a, Z_a) , se pueden hallar la longitud eclíptica geocéntrica λ_a y la latitud eclíptica geocéntrica β_a mediante las expresiones:

$$\tan \lambda_a = \frac{Y_a}{X_a} = \frac{0.0002148125}{(-0.0001301244)} = -1.650824806, \quad (8.2)$$

y hallando su arcotangente tenemos $\lambda_a = 121.2057122^\circ = 8\text{h } 4\text{m } 49.37\text{s}$.

$$\tan \beta_a = \frac{Z_a}{\sqrt{X_a^2 + Y_a^2}} = \frac{0.000040385}{\sqrt{(-0.0001301244)^2 + 0.0002148125^2}} = 0.17542672, \quad (8.3)$$

y hallando su arcotangente, sabiendo que $Z_a > 0$, tenemos $\beta_a = 9.134961201^\circ = 9^\circ 8' 5.86''$.

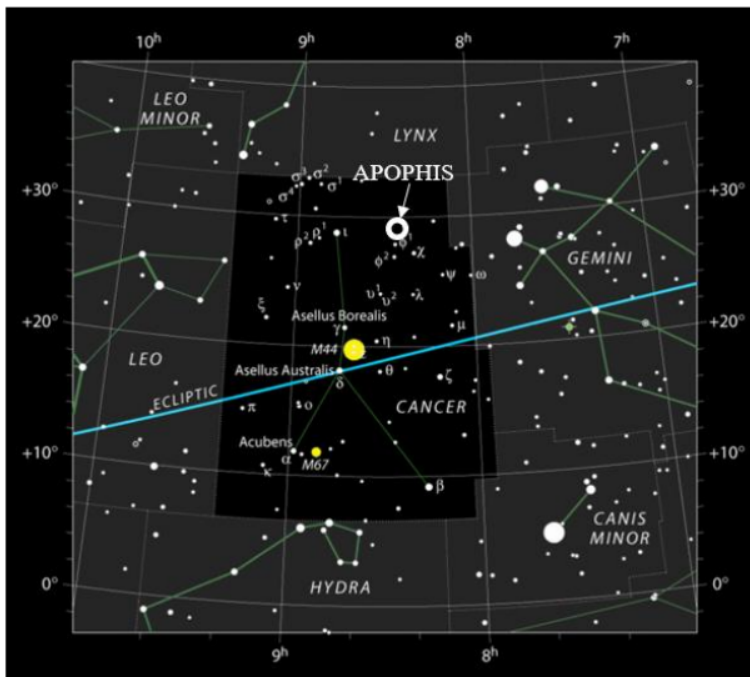


Figura 24. Localización de Apophis el 13 abril de 2029, a las 21h 44m 13s UT.

Coordenadas ecuatoriales (α_a, δ_a).

Conocidas las coordenadas eclípticas heliocéntricas, se pueden calcular sus coordenadas ecuatoriales (α_a, δ_a), mediante las siguientes expresiones, en las que ϵ representa la inclinación de la eclíptica en el año 2029, $\epsilon = 23.4360^\circ = 23^\circ 26' 9.6''$. Para la ascensión recta

$$\tan \alpha_a = \frac{\sin \lambda_a \cos \epsilon - \tan \beta_a \sin \epsilon}{\cos \lambda_a}. \quad (8.4)$$

Sustituyendo valores obtenemos $\tan \alpha_a = -1.391203158$, y sabiendo que el numerador es positivo, y hallando su arcotangente, tenemos que $\alpha_a = 125.708666396^\circ = 8\text{h } 22\text{m } 50.08\text{s}$.

Para la declinación δ_a :

$$\sin \delta_a = \sin \beta_a \cos \epsilon + \cos \beta_a \sin \epsilon \sin \lambda_a. \quad (8.5)$$

Sustituyendo valores y hallando su arccoseno tenemos que $\delta_a = 28.7852375201^\circ = 28^\circ 47' 6.86''$. Estas coordenadas ecuatoriales nos sitúan a Apophis el 13 de abril, cuando su máximo acercamiento a la Tierra, localizado entre la estrella ι de la constelación de Cáncer y Pollux, la estrella β de la constelación de Géminis.



Figura 25. Edificio Villanueva en el Real Observatorio Astronómico de Madrid [7].

Coordenadas horizontales (A, h) de Apophis para la latitud de Madrid.

Se ha elegido como lugar de observación el Real Observatorio Astronómico de Madrid, conocido como el Edificio Villanueva, cuyas coordenadas geográficas son

$$\text{Latitud, } \varphi = 40.40840278^\circ\text{N} = 40^\circ 24' 30.25''\text{N} \quad (8.6)$$

$$\text{Longitud, } \lambda = 3.68763611^\circ\text{O} = 3^\circ 41' 15.49''\text{O}. \quad (8.7)$$

Este complejo de edificios se encuentra situada sobre una colina cercana al Parque del Retiro, albergando hoy en día el Observatorio Astronómico Nacional y el Observatorio Geofísico Central, dependientes ambos del Instituto Geográfico Nacional.

Antes de abordar los cálculos de las Coordenadas Horizontales del asteroide, es necesario conocer el Tiempo Sidéreo local (θ_1), correspondiente al momento de observación en la fecha de estudio y en el lugar escogido. Entendemos el Tiempo Sidéreo como el ángulo horario H que forma el Punto Aries \Uparrow en un momento dado, con el meridiano del lugar, medido sobre el Ecuador celeste, desde el punto cardinal Sur y en sentido de las agujas del reloj (hacia el oeste). Este dato, sacado de efemeridesastronomicas.dyndns.org nos permitirá saber el ángulo horario de Apophis (H_a) haciendo uso de la siguiente expresión: $\theta_1 = H_a + \alpha_a$. Despejando, $H_a = \theta_1 - \alpha_a$, en la que α_a es la ascensión recta de Apophis, hallada anteriormente.

El tiempo sidéreo es $\theta_1 = 10\text{h } 58\text{m } 55.13\text{s} = 164.72970830^\circ$, y por tanto $H_a = \theta_1 - \alpha_a = 164.72970830^\circ - 125.7086664^\circ = 39.0210419036^\circ$. Conocidos estos datos, ya podemos hallar las coordenadas horizontales de Apophis sobre el cielo de Madrid, para lo cual nos serviremos de las siguientes ecuaciones:

Para el acimut A :

$$\tan A = \frac{\sin H_a}{\cos H_a \sin \varphi - \tan \delta \cos \varphi}. \quad (8.8)$$

Sustituyendo valores, hallando la arcotangente, y sabiendo que $\sin H_a > 0$, entonces $A = 82.287208^\circ$ (medido desde el Sur) y $A = 262.287208^\circ$ (si medimos el acimut desde el Norte).

Para la altura h :

$$\sin h = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos H_a. \quad (8.9)$$

Sustituyendo valores, y sabiendo que la altura siempre está comprendida en el intervalo $[-90^\circ, +90^\circ]$, después de hallar su arcoseno tenemos $h = 56.162217^\circ = 56^\circ 9' 43.98''$.

Concluimos, por tanto, que Apophis se podrá ver esa tarde-noche mirando hacia el punto cardinal Oeste, ocupando un punto en el cielo a media altura sobre el horizonte madrileño. Esto será posible siempre y cuando los altos edificios de Madrid no lo oculten. La solución sería alejarse de la gran ciudad un poco para poder verlo. Finalmente, dejar constancia que si los cálculos hubieran sido realizados respecto al equinoccio medio J2000.0, entonces las Coordenadas Horizontales hubieran sido $A = 263.072113^\circ$ y la altura $h = 56.892040^\circ$. Como se puede comprobar, la diferencia no es mucha.

9. Cuerpos y objetos celestes peligrosos para la Tierra

Cada vez que veo documentales en los que se nos presenta nuestro planeta Tierra tal y como se ve en el espacio exterior, tomo conciencia de lo insignificante que somos en el Universo. Esta observación tuvo lugar por primera vez en febrero de 1990, cuando la sonda espacial Voyager 1 abandonaba la órbita de Neptuno y se adentraba en los confines del Sistema Solar, momento en el que tomó la última foto de la Tierra, convirtiéndose en una imagen tan bien narrada por el astrónomo americano Carl Sagan, con su peculiar estilo, cuando decía:

Todos hemos vivido y vivimos hoy en día sobre una mota de polvo suspendida en un rayo de sol, nuestra Tierra. No es más que un pequeñísimo grano en una vasta arena cósmica, un pálido punto azul dentro de la oscuridad cósmica.

No se puede describir con mayor maestría nuestra posición en el espacio. Lo desvalidos y expuestos que estamos ante los peligros que nos pueden llegar del cielo y sentir la necesidad de poner en valor y preservar, todo lo que se pueda, este maravilloso planeta que habitamos y, hasta donde sabemos, el único en el que hay Vida.

Esta conciencia de vulnerabilidad ante objetos celestes que puedan impactar sobre la Tierra se ha visto

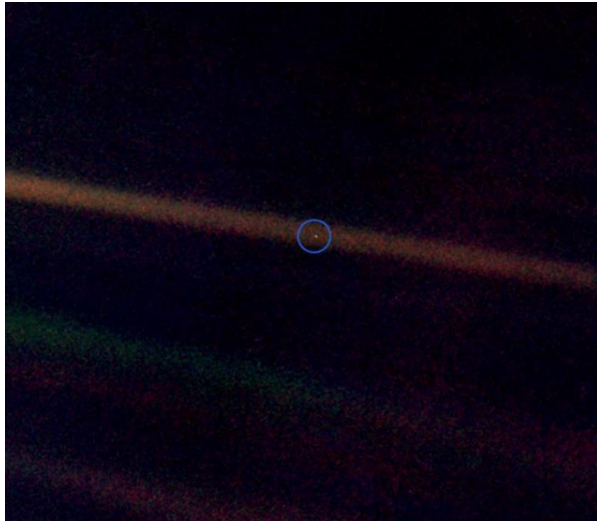


Figura 26. Fotografía de la Tierra tomada por la Voyager 1 en 1990. Crédito: NASA.

confirmada por episodios recientes, como los ya mencionados con anterioridad de Chelyabinsk (Rusia, 2013) y el Bólido de Tunguska (Rusia, 1908), o los fragmentos de meteoritos recogidos en el lago Tagish (Canadá, 2000), procedentes de un objeto mayor que explotó en la estratosfera.

Es en este contexto donde surge el concepto de NEO, unos cuerpos cuyas órbitas cruzan las de la Tierra, representando un peligro para ella. Hoy en día se conocen unos 34.000 NEOs, la mayoría de ellos asteroides de distintos tipos y tamaños, mientras que otros son cometas de corto periodo y pequeña inclinación, que no se alejan más allá de la órbita de Júpiter. Hacia alguno de estos NEOs ya se han enviado sondas espaciales para estudiarlos y valorar posibles métodos de interacción con ellos, para minimizar el riesgo que suponen para todos los habitantes del planeta.

En general, se puede decir que hay dos tipos de métodos de actuación. Uno es rápido y contundente y consistiría en el uso de una explosión controlada sobre el cuerpo celeste, o el choque contra él de una nave espacial, viajando a gran velocidad, para así intercambiar sus respectivos impulsos cinéticos y poder alterar su órbita. El otro es más lento en su aplicación, basándose en la generación de pequeños impulsos sobre el cuerpo para que, con el tiempo, varíe su órbita. La elección de uno u otro método debe ser muy meditada y estar basada en argumentos científicos lo más certeros posibles. Lo primero a realizar sería conocer, con la mayor precisión, la órbita de estos objetos, lo que implica medir, con el menor margen de error, sus elementos orbitales. Esto es difícil de conseguir, pues no es posible describir una órbita exacta para ninguno de ellos, pues existen una serie de variables aleatorias que no son predecibles. Se asume, por tanto, la existencia de una estrecha e incierta región del espacio por donde el objeto celeste va a pasar, conocida como *línea de variación* (LOV). En el caso de Apophis, esta región espacial tiene unas dimensiones de unos 3500 km de largo y 100 km de ancho. Esta región puede variar a causa de la posible resonancia 7:6 entre el asteroide y el planeta Júpiter, a lo que hay que sumar las perturbaciones que sufrirá su órbita después de su extremo acercamiento a la Tierra del 2029, haciendo que los dos cuerpos vuelvan a estar en la misma posición y con similar riesgo de colisión durante su paso del año 2036.

Una vez se haya conseguido una medición muy exacta de sus parámetros orbitales teniendo en cuenta estas variables, habría que definir qué se entiende como una órbita segura para Apophis, lo cual implicaría conseguir una desviación menor de 0.05 UA (~ 7000000 km) respecto al MOID, para así poder evitar futuras resonancias con otros cuerpos planetarios que pudieran volver a repetir el estado de peligrosidad

del asteroide en la actualidad. Lo segundo a valorar y tener en cuenta es el tiempo necesario para poder actuar sobre el asteroide o cometa, mediante el envío de una misión espacial hacia ellos. Hay que tener en cuenta que son muchos los NEOs conocidos, pero hay otros tantos objetos potencialmente peligrosos que se desconocen, y que pueden acercarse a gran velocidad a la Tierra, siendo invisibles por su tamaño y distancia a nosotros, haciendo imposible una actuación oportuna y ventajosa sobre los mismos, por falta de tiempo, que evitara las consecuencias catastróficas de un impacto sobre nuestro mundo. Se hace necesario programar con suficientes años de antelación un proyecto que minimice o erradique cualquier peligro que se presente en un momento dado, tanto fuera esperado como inesperado.

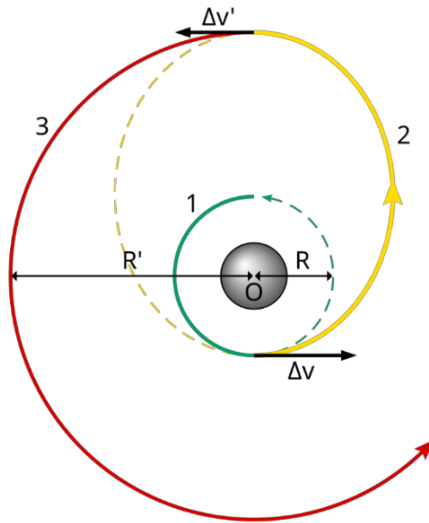


Figura 27. Órbita de transferencia de Hohmann [8].

Es por esto que sería aconsejable que estuviera elegido ya el lugar -o lugares- de lanzamiento de la misión y el momento temporal más adecuado para hacerlo, así como conocer la órbita de estacionamiento en donde colocar la sonda espacial y desde la cual realizar las maniobras de transferencia de trayectoria, siendo la de Hohmann (Fig. 27) la más adecuada, pues necesita una menor cantidad de energía en su desarrollo. La órbita de estacionamiento más adecuada es la geocéntrica circular (la 1 y verde en la figura), a unos 300 km de altura y moviéndose en un plano polar de 90° de inclinación respecto al ecuador. La órbita de transferencia (la 2 y amarilla) le aporta la energía necesaria para colocar la sonda en otra órbita circular más grande (la 3 y roja), que correspondería al sitio desde donde debería partir la misión de intercepción del objeto.

También podrían estar disponibles estas sondas para obtener información actualizada y rigurosa de las características, tanto materiales como dinámicas de estos NEOs. En el caso de Apophis, una sonda que se posara sobre su superficie aportaría información sobre su órbita y composición material. Por último, y aunque fuera difícil de organizar, dada la implicación de distintos Estados del mundo, sería conveniente contar con una proyección, sobre la superficie de la Tierra, de la trayectoria que va a seguir el asteroide, para así poder proteger del posible impacto a las personas que viven en esas zonas geográficas. Respecto a los métodos a utilizar, hay que cuantificar muy bien las ventajas y desventajas que conllevan cada uno de ellos. De este modo, el método de impacto cinético de una sonda contra el objeto para variar su trayectoria, se presenta como más seguro y eficaz a largo plazo, pero necesita de mayor tiempo en su preparación y lograr alcanzar la velocidad suficiente para que el impulso que transmita al objeto sea eficaz.

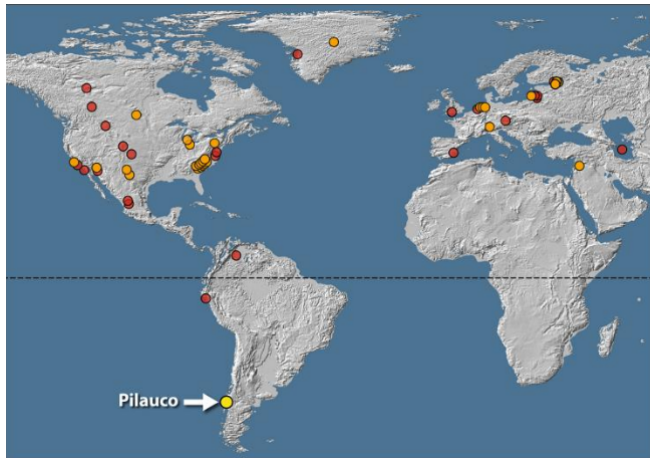


Figura 28. Mapa mostrando lugares con evidencias del impacto que produjo el Dryas Reciente. Crédito: Francis Thackeray (University of the Witwatersrand, Suráfrica).

El método de explosión de una carga nuclear resulta muy costoso de enviar, pero es bastante expeditivo en los resultados obtenidos. Por el contrario, tiene el inconveniente de que los fragmentos resultantes de la detonación siguieran en la trayectoria de colisión, con el resultado de provocar impactos múltiples sobre la superficie terrestre, con unas potenciales consecuencias aún más dramáticas para la población. No sería la primera vez que sucediera esto en la historia reciente de la Tierra, pues todo apunta a que el comienzo del último periodo glacial, conocido como Tardiglacial o Dryas Reciente, ocurrido hace unos 12900 años, que trajo unos cambios climáticos repentinos y drásticos que produjeron la extinción masiva de muchos seres vivos y la desaparición de algunas culturas humanas –como la Clovis en Norteamérica-, tuvo su origen en el impacto de múltiples escombros de asteroides, desgajados de un cuerpo mayor que se fragmentó y que chocaron contra la Tierra, principalmente en el Hemisferio Norte. Expuesto todo lo anterior, es el momento de recoger todas las misiones llevadas a cabo, hasta hoy, por parte de Instituciones especializadas como la NASA y ESA, o por países que están trabajando en este objetivo común de abortar cualquier peligro traído a la Tierra por parte de un objeto celeste con suficiente potencial destructor.

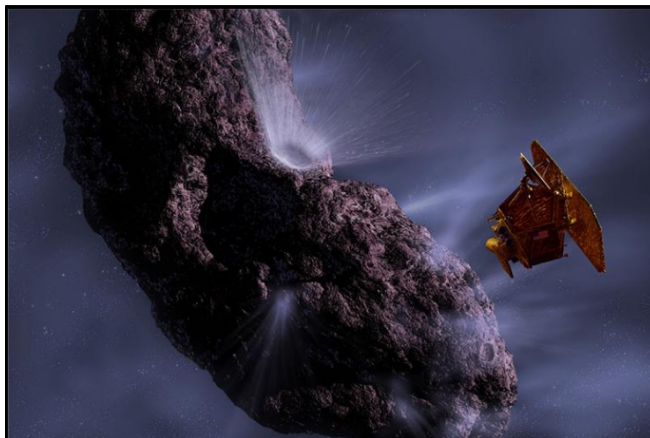


Figura 29. Recreación de la misión Deep Impact hasta el cometa Temple 1. Crédito: NASA.

10. Misiones espaciales para Defensa Planetaria

En esta apartado se van a recoger algunas de las más importantes misiones de Defensa Planetaria, llevadas a cabo en los últimos años por NASA, ESA y la Agencia Espacial japonesa, y las que están programadas en un futuro inmediato. El objetivo principal es ir acumulando experiencias y estrategias de actuación ante un peligroso objeto que se esté acercando a nosotros de forma amenazante. El trabajo consiste tanto en la obtención de información física y dinámica sobre estos NEOs, como en probar algunos métodos de actuación rápida ante un peligro inminente que se presente de manera imprevista. Así, ciertas misiones tuvieron o tienen como objetivo alcanzar algún asteroide o cometa conocido, a modo de banco de pruebas, mientras que otros se dirigen directamente hacia Apophis *el destructor*, por razones obvias de entender. Estas misiones expuestas en orden cronológico son:

10.1. Deep Impact (2005)

Esta misión fue lanzada por NASA el 12 de enero de 2005, con el objetivo de enviar una sonda, *el impactador*, de 327 kg de peso, portando un núcleo de cobre, de 100 kg, viajando a gran velocidad (10.2 km/s), hasta contactar con el cometa Temple 1, un cometa con unas dimensiones de 8×4 km. La misión llegó a su destino el 3 de julio de ese mismo año, separándose entonces en los dos cuerpos (el impactador y la nave de sobrevuelo) que conformaban la misión. El impacto sobre el asteroide ocurrió al día siguiente, 4 de julio, originando un cráter de unos 100 m de diámetro. El cráter fue fotografiado en el año 2011 por la nave espacial Stardust, enviada por la NASA en 1999 para recoger muestras de polvo del cometa Wild 2, y que pudieran ser enviadas por Stardust a la Tierra, para su análisis posterior. Una ampliación de la misión de esta nave fue acercarse al cometa Temple 1, donde tomó la fotografía del impacto.

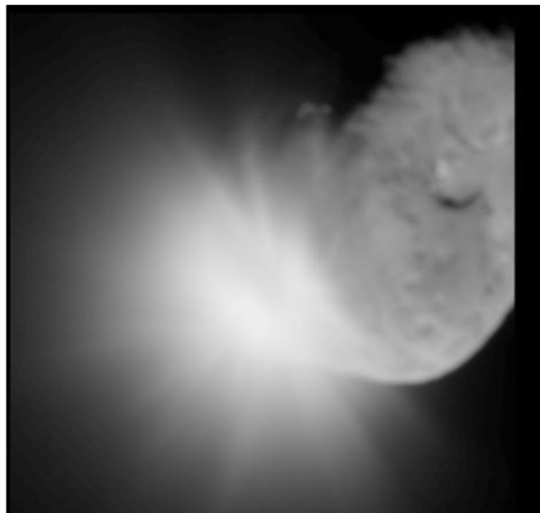


Figura 30. Fotografía del impacto en el cometa Temple 1, provocado por la misión Deep Impact. Crédito: NASA/JPL-Caltech/UMD.

El choque eyectó al espacio materiales del cometa y polvo que fueron analizados por los científicos, determinando la naturaleza porosa de Temple 1 y la existencia de materiales orgánicos presentes en la pluma brillante que originó la explosión en su superficie. Los científicos determinaron también que el cometa, de la familia cometaria de Júpiter, se había formado en la Nube de Oort. Como resultado sobre la dinámica del cometa se observó un cambio en su velocidad, en una cantidad de 0.0001 m/s, y la teórica

reducción de su distancia de perihelio en unos 10 m. Deep Impact, fue utilizada después en el estudio de otros cometas, como el Hartley 2 y Boethin, llegando a este último pero sin conseguir el objetivo planteado. Posteriormente su misión ha sido completada con la búsqueda de planetas extrasolares en algunas estrellas. En el año 2013 su señal se perdió definitivamente, dejando de enviar información.

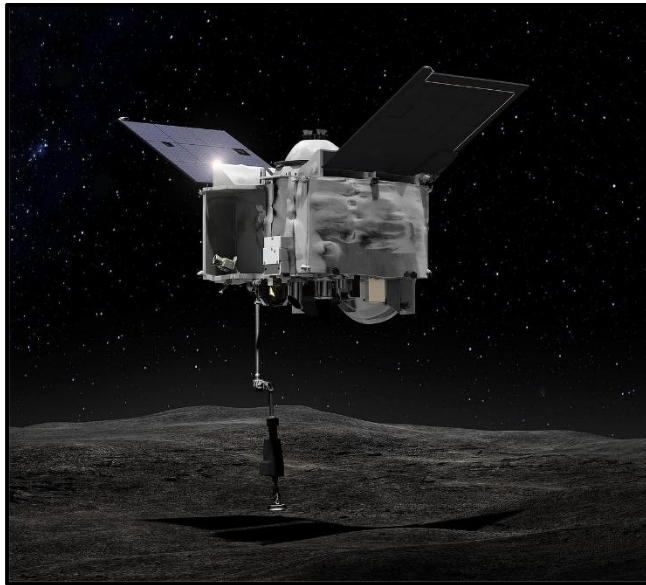


Figura 31. Osiris-Rex recogiendo una muestra de materia del asteroide Bennu. Crédito: NASA.

10.2. Osiris-Rex (2016)

Esta misión de NASA fue lanzada en 2016 con el objetivo de recoger una muestra de la superficie del asteroide Bennu, mediante un brazo robótico. Esta muestra, tomada en 2020, confirmó cuando fue analizada en 2023 por los científicos, que era un asteroide carbonoso con moléculas orgánicas en su composición. Fue la primera misión de NASA que devolvió muestras extraídas de un asteroide. Una nave similar a Stardust fue la encargada de traer a la Tierra la muestra recogida, la cual fue recibida en el desierto de Utah, usando un paracaídas en su aterrizaje. Otra de las misiones que llevaba Osiris-Rex era analizar y medir los efectos no gravitacionales (llamado efecto Yarkovsky) producidos por la rotación de un cuerpo con forma irregular en el espacio, cuando absorbe la radiación solar en una parte del asteroide y la irradia al espacio por otra parte de forma diferencial a la absorbida, alterando de esta manera su órbita.

En el año 2022, se decidió ampliar la misión de Osiris-Rex, dirigiéndola hacia Apophis, rebautizándola como Osiris Apophis Explorer. Antes de acometer la misión, tuvo que darseis vueltas alrededor del Sol y otras tres más alrededor de la Tierra, para esperar el momento más propicio y aumentar su energía cinética. La misión se encontrará con el asteroide a principios del mes de abril de 2029, un poco antes de su máximo acercamiento a la Tierra, comenzando a grabar imágenes de Apophis a medida que se acerca a él.

Llegará al asteroide a finales de abril y permanecerá dieciocho meses orbitando alrededor del mismo, de una manera similar a lo hecho en Bennu, para tomar imágenes de todas sus caras, medir su rotación y, finalmente, haciendo uso de sus propulsores llegará a situarse a cinco metros de su superficie, la cual

removerá, para poder analizar espectralmente el suelo y el subsuelo de Apophis. También estudiará cómo cambió su superficie después de su interacción con el fuerte campo gravitatorio de la Tierra, ya que se espera que ocurran pequeños terremotos y deslizamientos de tierra en su superficie.

10.3. Dart (2021)

Estas siglas significan *Demonstration for Autonomous Rendezvous Technology*, convirtiéndose así en la primera prueba encaminada a lograr un cambio en la órbita de un asteroide, mediante el impacto sobre el mismo de una nave espacial, lográndose una transferencia mutua del impulso entre ambos cuerpos después de la colisión. La nave fue lanzada en noviembre de 2021 por NASA, fijándose como objetivo a alcanzar al sistema binario de asteroides Didymos (780 m de diámetro) y su luna Dimorphos (160 m de diámetro). La misión constaba de dos ingenios espaciales. La nave principal era el cuerpo impactante (610 kg), sin carga científica útil, pues sólo portaba sensores para la navegación. La nave secundaria, llamada LiciaCube, italiana de 1.33 kg y 10 cm de arista, se separó de ella quince días antes del impacto con la intención de tomar imágenes del mismo y de la posterior eyección de materia al espacio.

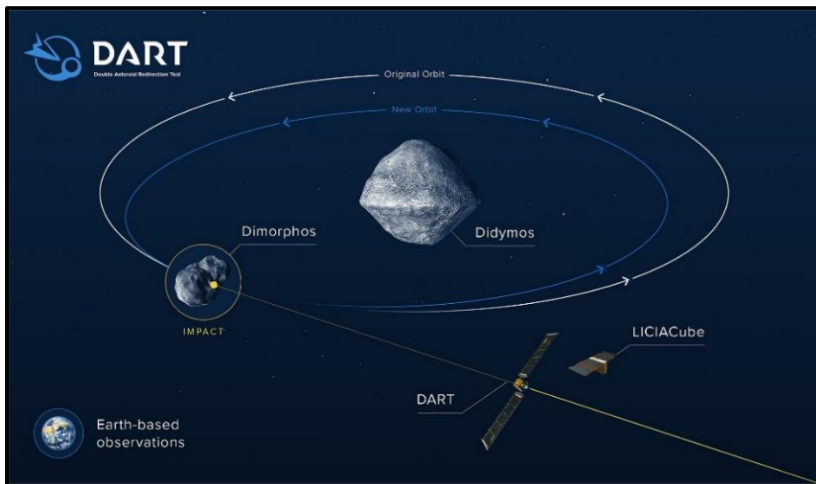


Figura 32. Misión DART de la NASA hacia el sistema binario de asteroides Didymos-Dimorphos. Crédito: NASA/John Hopkins.

La colisión tuvo lugar el 26 de septiembre de 2022, cuando la nave espacial, viajando a una velocidad de 6.6 km/s, originó en Dimorphos un Δv de unos 0.4 mm/s, una pequeña cantidad pero que, con el paso del tiempo, se acumularía hasta hacer variar su periodo de traslación alrededor de Didymos en 32 minutos, pasando de 11h 55m a otro menor de 11h y 23m. Este experimento no supuso ningún riesgo para la Tierra, pues este sistema binario de asteroides no cruza la órbita terrestre y, por el contrario, presentaba la ventaja de estar en el año 2022 cerca de nosotros, a sólo 0.075 UA. Dart supuso un hito importante en el programa de Seguridad Planetaria pues, a diferencia de Deep Impact - que fue un choque intencionado de una sonda espacial para estudiar la composición y estructura de un cometa - en esta ocasión, con DART, se cambió de forma deliberada la trayectoria de un objeto celeste, sirviendo como prueba de que se puede alterar su trayectoria y, llegado el caso, servir como modelo de actuación a seguir en el futuro ante un teórico impacto sobre la Tierra de un cometa o asteroide con efectos devastadores para la misma.

Con objeto de hacer un seguimiento de los resultados obtenidos con DART, pues existe algo de incertidumbre sobre los cambios en la velocidad real y en la órbita de Dimorphos, la ESA europea

está desarrollando una nave espacial, HERA, que será lanzada en 2024 para ver el tamaño del cráter originado por el impacto y medir el impulso adquirido por dicho asteroide.

10.4. HERA (2024)

Hera es una misión de la Agencia Espacial Europea. Tiene una masa de 1128 kilogramos y lleva una carga útil de cámaras, un altímetro y un espectrómetro. También llevará dos nanosatélites CubeSats, llamados Milani y Juventas, que serán liberados por HERA antes de llegar al sistema. Aunque es una misión de Defensa Planetaria, Hera abordará algunos retos científicos distintos a las anteriores misiones. Será la primera vez que se estudie la estructura y materia de un sistema binario de asteroides (Didymo-Dimorphos) además de su comportamiento dinámico mutuo. Se investigará la geofísica de la superficie de dos objetos que tienen diferente tamaño y gravedad, pero que se han formado en el mismo cuerpo original primario.



Figura 33. Misión HERA hacia el sistema binario de asteroides Didymo-Dimorphos. Crédito: ESA.

Los dos satélites tipo Cubesat tendrán diferentes misiones. Así, Juventas, que llevará un radar de baja frecuencia, debe realizar mediciones del subsuelo y de la estructura interna, contribuir a la determinación del campo gravitacional y proporcionar información sobre la respuesta mecánica de la superficie al aterrizar en la superficie de Dimorphos. Su compañera Milani recopilará datos espectrales de la superficie de los dos asteroides (composición de la superficie) e identificará la presencia de polvo en el espacio circundante. HERA será lanzada en octubre de 2024. Después de una breve estancia gravitacional en Marte, en marzo de 2025, observando la luna marciana Deimos, la nave espacial llegará al asteroide binario Didymos-Dimorphos el 28 de diciembre de 2026, cuatro años después de DART, para dar comienzo entonces a seis meses de investigación.

10.5. RAMSES (2027)

RAMSES (Rapid Apophis Mission for Security and Safety) es una misión espacial de ESA que tiene como objetivo estudiar el asteroide Apophis, antes y durante su acercamiento a la Tierra, en 2029, acompañándolo en su viaje para ver cómo se deforma y cambia su superficie debido a la fuerza de gravedad ejercida por nuestro planeta. La nave debería ser lanzada en los primeros meses de 2028, para poder llegar a Apophis en febrero de 2029. De esta manera, cuando Apophis esté volando sobre nuestras cabezas, en abril de 2029, habrá dos misiones espaciales acompañándolo para poder entender

mejor como se comportan los asteroides ante un campo gravitatorio planetario fuerte y conocer sus características físicas y materiales.

Queremos y debemos conocer muy bien todas estas propiedades, para que alguna vez seamos capaces de desviar un asteroide peligroso de su trayectoria de colisión con la Tierra. Si el cine funciona como premonición de ciertos acontecimientos, difícilmente imaginables, como sucedió con *Viaje a la Luna* (1902), de Julio Verne (en la época pionera del cine mudo), y luego esto se convirtió en realidad con la llegada a la Luna del hombre, el 20 de julio de 1969, con la misión estadounidense de Apolo XI; ahora la cosa es muy distinta y no queremos que las películas que narran impactos de cometas y asteroides asesinos sobre la Tierra se conviertan en una terrible realidad.



Figura 34. Misión RAMSES de ESA hacia el asteroide Apophis. Crédito: ESA.

Fuentes documentales online

- NASA Science. *NASA analysis: Earth is safe from asteroid Apophis for 100 plus years*, JPL, marzo 2021.
- Jet Propulsion Laboratory. *99942 Apophis (2004 MN4). Predicting Apophis' Earth encounters in 2029 and 2036*, 2013.
- Jet Propulsion Laboratory. *JPL Horizons online solar system data and ephemeris* (App)
- The Sky Live. *3D Solar System Viewer*.

References

- [1] <https://www.facebook.com/sociedad.astronomia/posts/cuándo-pudiera-ocurrir-el-próximo-avistamiento-de-un-enorme-meteoro-tipo-chelyab/4661557217208870/>
- [2] <https://istoedinheiro.com.br/asteroide-maior-que-o-empire-state-building-se-dirige-para-a-terra-diz-a-nasa/>
- [3] Creative Commons Attribution-Share Alike 3.0 Unported license. <https://creativecommons.org/licenses/by-sa/3.0/deed.en>

- [4] Creative Commons CC0 1.0 Universal Public Domain Dedication.
<https://creativecommons.org/publicdomain/zero/1.0/deed.en>
- [5] Sormano Astronomical Observatory (A. Testa). <http://www.brera.mi.astro.it/sormano/>
- [6] Initiative for Interstellar Studies. <https://i4is.org/sailing-to-apophis/#gsc.tab=0>
- [7] Creative Commons Attribution-Share Alike 4.0 Unported license.
<https://creativecommons.org/licenses/by-sa/4.0/deed.en>
- [8] Creative Commons Attribution-Share Alike 2.5 Generic license.
<https://creativecommons.org/licenses/by-sa/2.5/deed.en>
- [9] *Astronomical algorithms*, Jean Meeus (Editorial Willmann Bell, Richmond, Virginia, 2009).
- [10] *More Mathematical Astronomy, Morsels*, Jean Meeus (Editorial Willmann Bell, Richmond, Virginia, 2002).
- [11] *The Tunguska event*, Chris Trayner, Journal of British Astronomical Association **107** (1997).
- [12] *Rediseño de misión especial para desviar el asteroide Apophis de un encuentro cercano a la Tierra*, R. Fernández Frittelli, M. Jandar Paz, Walkiria Schulz, Universidad de Córdoba (Argentina), 2020.
- [13] *Asteroid (99942) Apophis will approach the Earth on April 13, 2029*, Francesco Manca, Piero Sicoli, Augusto Testa. Observatorio Astronómico Sormano, 2021.
- [14] *Sailing to Apophis LPSC-2016*, W. Paul Blasé, Adam Hibberd, Marshall Eubanks, Robert G. Kennedy III. European Space Agency (ESA).

SECTION: HERRAMIENTAS DEL OBSERVATORIO VIRTUAL

Caracterización de cúmulos estelares abiertos

Joaquín Álvaro Contreras¹¹FAAE, Madrid, Spain. E-mail: jalvaro@citelan.es.

Keywords: cúmulos estelares, herramientas VO, TOPCAT, Aladin, observatorio virtual, Gaia
© Este artículo está protegido bajo una licencia Creative Commons Attribution 4.0 License

Resumen

El problema de discriminar qué estrellas forman parte de un cúmulo estelar abierto, en adelante **OC** (*Open Cluster*), descartando aquellas que no son parte del cúmulo, es fundamental para determinar la existencia de un OC, y en definitiva caracterizarlo, atribuyéndole una estimación precisa de sus componentes y a partir de estos, la masa del cúmulo, edad, distancia, movimiento propio relativo en la galaxia, composición química, etc. El proceso de caracterizar un OC no es fácil y, aunque ya se utilizan técnicas de inteligencia artificial eficaces, conocer los fundamentos y tareas para resolver este problema es conveniente si se desea trabajar en este campo, ya sea para refinar datos de OCs conocidos como para descubrir otros que aún no lo son. Afortunadamente hay algunas herramientas del Observatorio Virtual (VO) que facilitan la tarea. En este artículo, y los siguientes de esta serie, haremos un repaso amplio sobre algunos casos prácticos utilizando herramientas VO como TOPCAT [1], Aladin [2], Clusterix 2.0 [3] y VOSA [4], e intentaremos finalmente un desarrollo con técnicas de inteligencia artificial con este mismo propósito.

Abstract

The problem of identifying which stars belong to an open star cluster (hereinafter OC, Open Cluster), while discarding those that do not, is fundamental to determining the existence of an OC and, ultimately, characterizing it. This involves providing an accurate estimate of its members and, based on them, calculating properties such as the cluster's mass, age, distance, relative proper motion within the galaxy, chemical composition, and more. The process of characterizing an OC is not straightforward, and although effective artificial intelligence techniques are already in use, understanding the fundamentals and tasks required to address this problem is essential for those who wish to work in this field, whether to refine data for known OCs or to discover previously unidentified ones. Fortunately, there are tools from the Virtual Observatory (VO) that make this task easier. In this article, and in subsequent ones in this series, we will provide a comprehensive review of practical cases using VO tools such as TOPCAT [1], Aladin [2], Clusterix 2.0 [3] and VOSA [4], and we will conclude by attempting a development using artificial intelligence techniques for the same purpose.

1. Introducción

Los cúmulos estelares abiertos son grupos de estrellas unidas gravitacionalmente que se formaron a partir de una misma nube de gas molecular, por lo que comparten la misma composición química y edad, y tienen además posiciones relativas muy próximas y movimientos propios similares. Los OCs son objetos fundamentales para comprender la estructura espiral, la dinámica y la evolución química de la galaxia, al tiempo que también ayudan a comprender los procesos de formación estelar.

Aunque la mayoría de las estrellas de la Vía Láctea se presentan aisladas, se considera que en su mayor parte, tal vez todas, se forman en entornos agrupados y pasan un periodo de tiempo unidas gravitacionalmente entre sí incrustadas en la nube progenitora, (*Clarke et al. 2000* [5]; *Portegies Zwart et al. 2010* [6]). La evolución de estos sistemas tiende a disgregarlos en pocos millones de años debido

a interacciones gravitatorias con otros sistemas, a fuerzas de marea galácticas y/o a mecanismos que involucran la pérdida de gas por retroalimentación estelar. Una fracción de estos sistemas sobrevive, no obstante, en la fase incrustada y permanece ligada a escalas de tiempo más largas.

Los OCs jóvenes permiten investigar las regiones de formación de estrellas y comprender los mecanismos que dan lugar a éstas. Por su parte los cúmulos intermedios y antiguos informan sobre los procesos estelares y la evolución del disco galáctico. Algunas órbitas muy perturbadas podrían también proporcionar evidencia de sucesos recientes de fusión y rastros de acreción pasada desde fuera de la galaxia, (*Cantat-Gaudin et al. 2018 [7]*).

Los OCs pueden contener desde sólo unas decenas de componentes a unos pocos miles de estrellas y suelen encontrarse casi exclusivamente en el plano de la galaxia.

El estudio de los OCs ha tomado nuevo impulso a partir del inmenso y preciso conjunto de datos de la misión GAIA (ESA) [8], tanto para la revisión de los ya conocidos como para la determinación de otros nuevos, lo que ha aumentado el catálogo de OCs conocido hasta fechas recientes a prácticamente doblarlo con unos 3.800 en el momento actual [9].

2. Caracterización de cúmulos estelares abiertos

En este primer artículo analizaremos, como caso práctico, el cúmulo **NGC 2682 (Messier 67)** [10], bien conocido, lo que nos permitirá un test sencillo de los resultados obtenidos. Para éste, y otros ejercicios similares, puede tomarse como referencia de datos ya catalogados el *CDS VizieR Catalogue of Open Clusters* [11], del que sólo haremos uso para tomar una estimación inicial del campo de cielo a estudiar y, al final, para comparar resultados con los datos aquí recogidos.

Antes de empezar podemos ver las dos imágenes presentadas en la Fig. 1. A la izquierda se muestran, (*como puntos azules*) todas las estrellas presentes en el campo de visión seleccionado¹, donde se presume que está localizado NGC 2682. A la derecha, (*como puntos amarillos*), se resaltan las estrellas que forman parte del cúmulo. Esto nos da idea de cuál es el objetivo perseguido, la caracterización del cúmulo, que empieza por determinar las estrellas que forman parte del mismo.

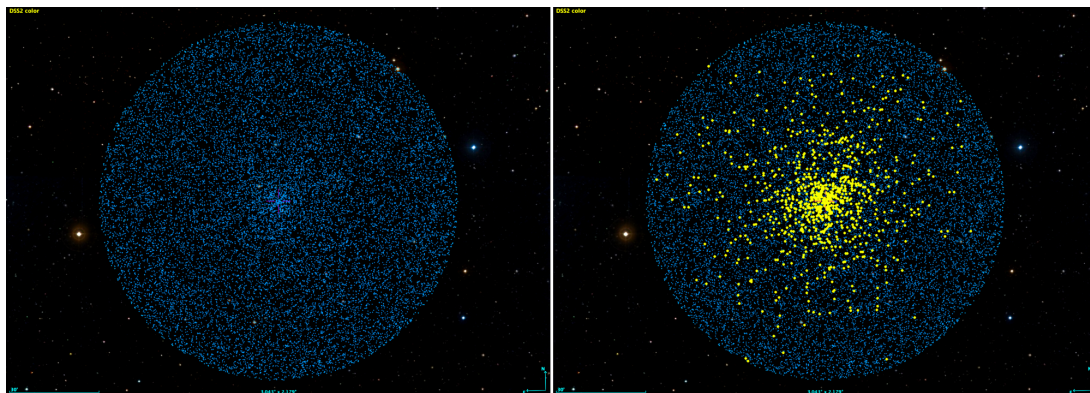


Figura 1. Izq: 20.284 estrellas a partir de Gaia/EDR3, (ver nota al pie). Dcha: sobreimpresas y resaltadas en amarillo 920 estrellas estimadas como componentes de NGC 2682 a partir de un análisis rápido con TOPCAT. Ambas imágenes son representaciones en *Aladin Sky Atlas* con los datos transferidos desde TOPCAT.

¹Obtenidas de Gaia/EDR3 con un campo de cielo de 1° para la posición estimada del centro del cúmulo en coordenadas ecuatoriales. En total aparecen 20.284 estrellas hasta la magnitud visual 22,26

Para el propósito que nos ocupa en este artículo se utilizará **TOPCAT** y, opcionalmente **Aladin** como atlas del cielo, con el único objetivo de tener una representación visual ilustrativa. Si se desea reproducir el hilo de procesos que vamos a seguir (*deseable*), es necesario instalar ambas herramientas VO, (ver referencias [1] y [2]; en ambos casos hay instrucciones para la instalación y tutoriales de las dos aplicaciones, si se desea profundizar en ellas. También el *Observatorio Virtual Español (SVO)* tiene publicadas varias escuelas y talleres al respecto) [12], (ver caso práctico del SVO, por Francisco Jiménez-Esteban y Belén López Martí [13]).

El análisis de un campo estelar que presumiblemente contiene un OC, con objeto de caracterizarlo, inicialmente puede hacerse a partir de algunas características esbozadas en la definición de 'cúmulo estelar abierto': *conjunto de estrellas nacidas de una misma nube de gas molecular en una ventana temporal relativamente corta*. Esto significa que las estrellas del cúmulo van a compartir una serie de propiedades. A saber:

1. La dinámica de la nube de gas progenitora. Esto indica que estas estrellas presentarán movimientos propios similares, (tanto en ascensión recta como en declinación), frente al conjunto de estrellas del campo de visión que, en general, tendrán una distribución aleatoria de velocidades.
2. Además serán estrellas próximas entre sí. Es decir, su distancia a un punto de referencia, como puede ser la Tierra, debe ser la misma, dentro de un rango más o menos reducido debido a la propia extensión del cúmulo.
3. Por último, estas estrellas tendrán una misma composición química, (heredada de la nube original), y una misma edad, también dentro de un rango temporal bastante estrecho.

3. Primeros pasos - Obtención de datos y análisis sobre velocidades

Empezaremos por cargar en TOPCAT, desde Gaia/EDR3, datos estelares del campo de cielo deseado, para iniciar seguidamente un primer filtrado atendiendo a los movimientos propios, según el punto 1.

3.1. Acceso a datos en Gaia desde TOPCAT

Se ejecuta la aplicación TOPCAT que se iniciará presentando una ventana simple sin datos ni opciones de menú activas ya que aún no se ha cargado ninguna tabla. La única opción posible es la de 'cargar una nueva tabla'. Seguiremos la secuencia de pasos indicados en la Fig. 2 y que se detallan a continuación.

1. cargar nueva tabla - Se abre una segunda ventana donde seleccionaremos el icono marcado con el número 2 'obtener catálogos de fuentes usando los servicios web de búsqueda por conos'. (Los menús son contextuales)
2. al seleccionar el ítem marcado con el número 2 se abre una tercera ventana.
3. en *Keywords* escribimos 'gaia' y,
4. pulsamos en 'encontrar servicios'
5. en la lista de recursos que se despliega, seleccionamos 'GAIA DR3' o 'GAIA EDR3'
6. ahora indicamos el nombre del objeto que nos interesa. En este caso 'NGC 2682' y,
7. pedimos que lo resuelva. En este paso se rellenarán automáticamente las coordenadas (RA, Dec) del objeto buscado. (También es posible escribir directamente estas coordenadas, si se conocen, en los campos correspondientes).
8. ya sólo falta indicar el radio del campo de cielo a considerar. En este caso elegimos 1º (60 minutos de arco)²
9. una vez completados estos pasos, confirmamos la búsqueda pulsando 'OK'

²De los datos conocidos de este cúmulo -ver referencia [7]- sabemos que su diámetro estimado es de 25 arcmin. Aquí seremos algo más generosos con el tamaño del campo a analizar.

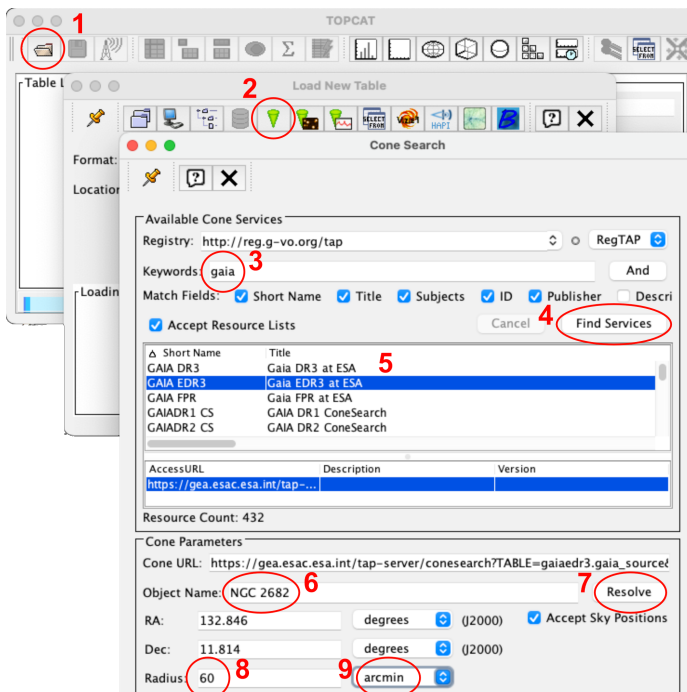


Figura 2. Captura de pantalla de tres ventanas consecutivas en TOPCAT. (Las ventanas se van desplegando a medida que se pulsán las acciones 1, 2, ...) Los números correlativos indican los pasos sucesivos a seguir para obtener los datos de Gaia para NGC 2682. Los detalles en cada caso se explican en el texto del artículo.

En pocos segundos se habrá cargado la tabla correspondiente con los datos extraídos de Gaia y la ventana principal de TOPCAT nos dará información de su contenido (Fig. 3).

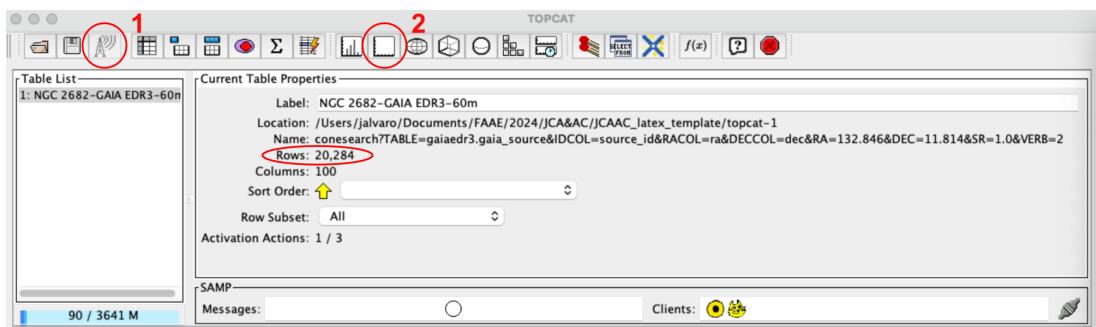


Figura 3. 20.284 registros recuperados de Gaia con los datos estelares disponibles para esta región del cielo.

Aquí podemos explorar esta información, tanto los datos brutos de la tabla como los metadatos de la misma o las columnas que la componen. Las opciones disponibles están accesibles desde los iconos contextuales en la parte superior de esta ventana.

Por ahora nos limitaremos a dos de estas funcionalidades. La primera, opcional, será exportar estos datos a *Aladin*. Para ello es preciso abrir esta aplicación (*Aladin*) y escribir en la línea superior de comando: NGC 2682 (+INTRO), con objeto de situar en el atlas del cielo una posición y escala aproximada. Hecho esto podemos pulsar el icono correspondiente en TOPCAT, (marcado con '1' en la Fig. 3). El resultado será similar a lo presentado en la Fig. 1 (izquierda).

Pero lo que nos interesa para continuar en nuestro objetivo de caracterizar el cúmulo es ver, sin más

preámbulos, la información disponible en el espacio de configuración de velocidades, (movimientos propios en AR y Dec).

3.2. Análisis de movimientos propios en las estrellas del campo seleccionado

Empezaremos por representar el mapa de movimientos propios del *dataset* descargado. Para ello pulsamos en el icono superior 'ventana de trazado de mapas', (marcado con '2' en la Fig. 3). Se abrirá una nueva ventana en la que debemos seleccionar los parámetros que queremos representar, en este caso movimiento propio en ascensión recta y movimiento propio en declinación, (*pmra* y *pmdec*, del conjunto de datos). Ver Fig.4 - izquierda.

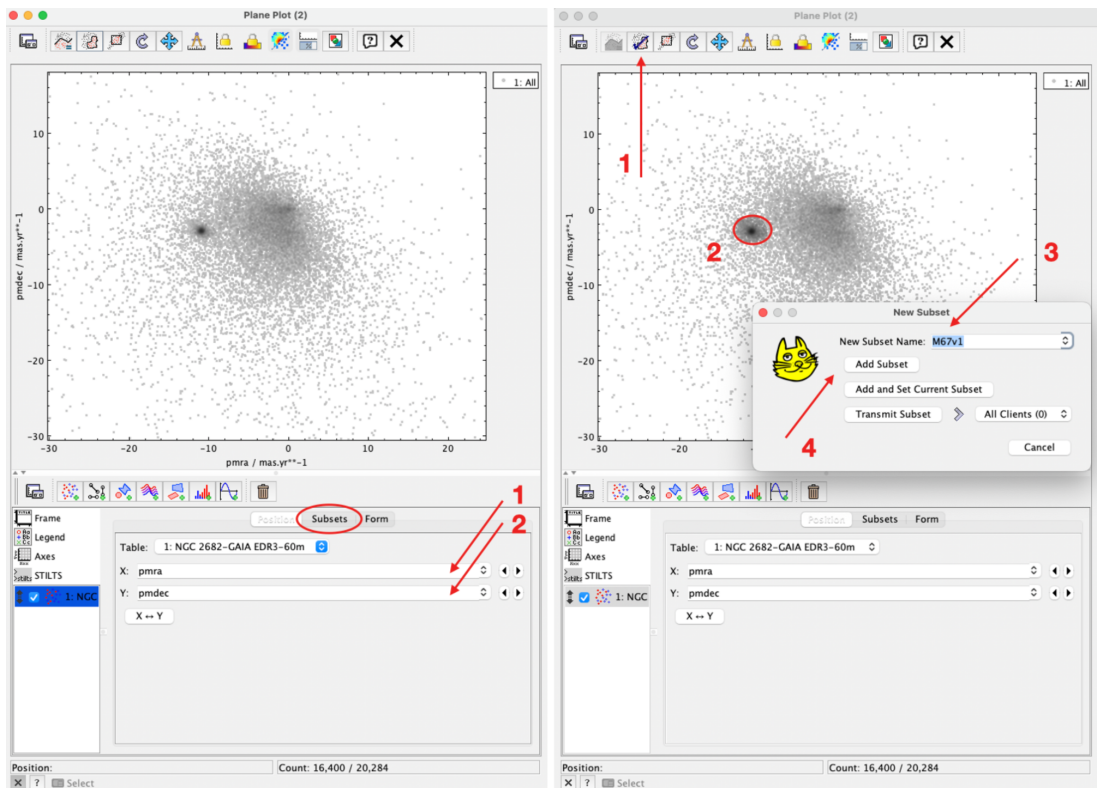


Figura 4. Izquierda: mapa de velocidades del conjunto de estrellas descargado de Gaia. Derecha: selección de candidatos a formar parte del cúmulo. En la parte inferior de ambas imágenes se puede ver que del total de 20.284 estrellas extraídas de Gaia, sólo 16400 han sido representadas. El resto corresponden a estrellas que no tienen informada alguna de las dos velocidades o ninguna de ellas.

Para obtener el gráfico de movimientos propios, (Fig.4 - izquierda), se debe seleccionar el parámetro 'pmra' en el eje X y el parámetro 'pmdec' en el eje Y, en las posiciones marcadas con '1' y '2' en la figura. La representación será inmediata. Se puede ajustar la escala de ambos ejes en el mapa usando el ratón en la manera habitual para ajustar el 'zoom'. También se puede elegir el color con el que se presentan los puntos de este mapa a partir de la pestaña 'subsets' marcada en la imagen.

Lo que podemos ver en este gráfico es que las velocidades de las estrellas se distribuyen en general con cierto carácter aleatorio presentando una mayor abundancia en torno a los '0' en ambos ejes. Pero se

puede observar también una *sobre-densidad*, muy evidente y concentrada, en torno a $pmra -10$ y $pmdec -2$. Esto es justamente lo que buscábamos: una concentración de estrellas con movimientos propios muy similares y definidos, como ésta, revela con toda seguridad la existencia de un OC.

El siguiente paso es seleccionar el conjunto de puntos de la sobre-densidad para crear un *subset*. Asumimos que con ello estamos creando un filtro que separará los posibles componentes del cúmulo del resto de estrellas de fondo. Para ello veamos ahora la parte derecha de la Fig. 4:

Empezaremos por seleccionar el icono correspondiente a '*dibujar una región a mano alzada para definir un nuevo subset*', (paso número '1' de la Fig. 4 -derecha-). Utilizaremos el ratón para encerrar la región señalada con un bucle en forma de lazo, (punto '2' Fig. 4 -derecha-). Una vez cerrado el lazo se debe volver a *pinchar* el mismo icono de antes ('1') y en la ventana emergente daremos nombre a este subset (punto '3'). Conviene añadir un número a estos nombres porque serán varios los filtros sucesivos que aplicaremos. Hecho esto, pulsaremos el botón '*add subset*' (paso '4').

Después de estas primeras acciones tenemos un subconjunto del total al que hemos llamado '*M67v1*' que contiene 2.011 estrellas (ver remarque en la Fig. 5) y que es resultado de la selección hecha casi *artesanalmente* a mano alzada sobre el mapa inicial.

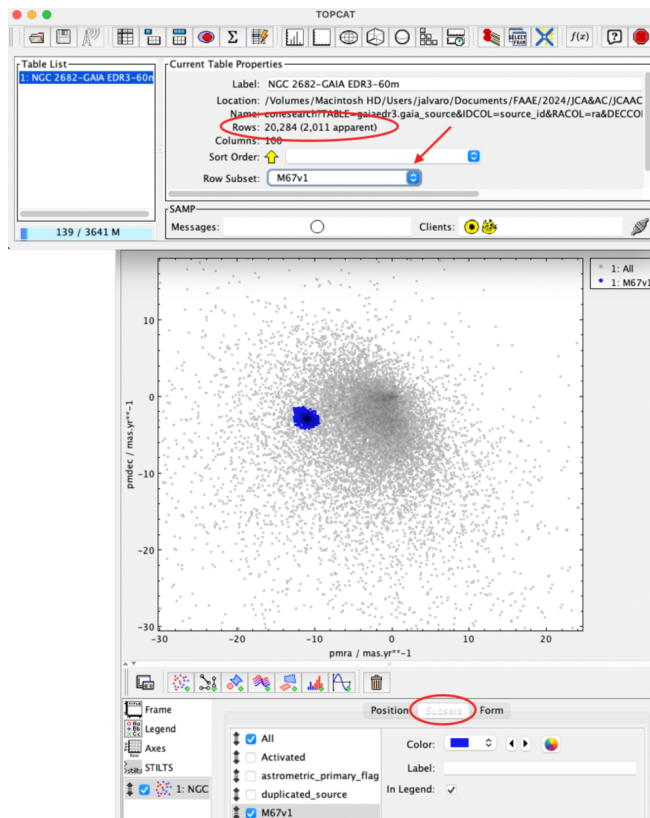


Figura 5. Representación del subset resultado del primer filtrado (en azul), sobre el fondo del dataset total inicial. En la ventana superior se puede elegir entre los datos del dataset original 'all' y cualquiera de los subsets que se irán creando -combo marcado con la flecha-. En la ventana inferior, la pestaña 'subsets' permite activar/desactivar los diferentes conjuntos/susconjuntos representados, así como otras propiedades gráficas de los mismos, como el color.

El hecho de ir paso a paso enumerando con detalle cada una de las acciones a realizar puede distraernos de alguna manera del propósito real, que es ir progresando en el proceso de discriminar los componentes del OC. Se ha optado por ser minucioso en lo posible en esta primera fase con objeto de facilitar la tarea de seguir este caso práctico a las personas que afrontan por primera vez el trabajo con TOPCAT. A medida que vayamos avanzando no será necesario ser tan prolijos en detalles y el hilo conceptual irá ganando peso.

Lo que debe quedar claro después de esta primera selección es que, a todas luces, es insuficiente para dar por resuelto el problema. Se trata de una primera aproximación sobre la que se debe seguir trabajando. Lo siguiente que haremos es aplicar un nuevo filtro sobre el subset 'M67v1'. Para ello acotaremos algo la posible dispersión de las velocidades en 'M67v1' que, como hemos visto, se ha seleccionado un poco 'a ojo' y en el que, por precaución, se ha sido algo generoso (por exceso) a la hora de *dibujar* la región de interés en el mapa del espacio de configuración de velocidades. También aprovecharemos el nuevo filtro para descartar aquellas estrellas en las que el error en los datos de movimientos propios sea mayor de lo razonable.

Hacemos uso ahora de otra funcionalidad de TOPCAT que nos permite tener información resumizada y estadística de los datos que manejamos.

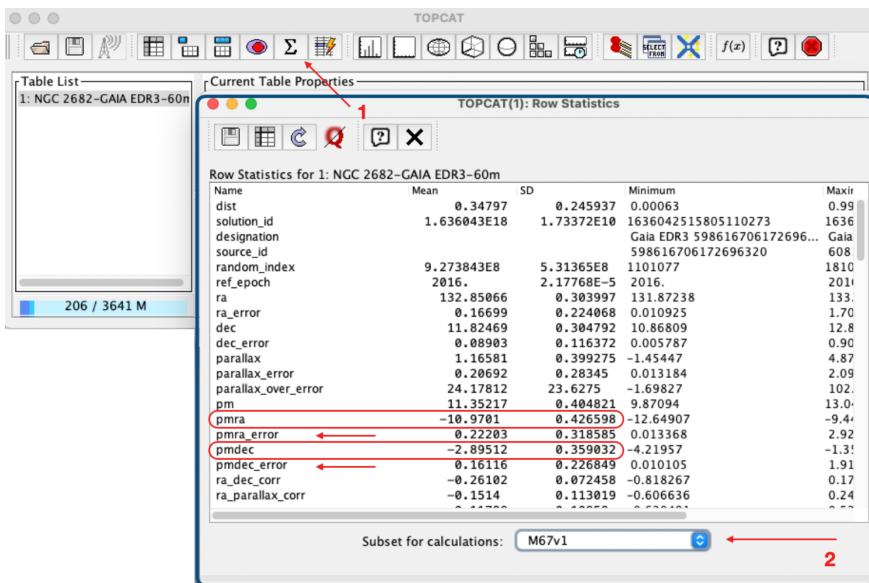


Figura 6. Detalle de valores estadísticos para cada columna de la tabla. Datos referidos al subset 'M67v1'. Remarcados en la figura los campos significativos en esta etapa del trabajo. Las indicaciones correspondientes se detallan en el texto del artículo.

En la ventana inicial de TOPCAT seleccionamos ahora la función de '*mostrar estadísticas*', (paso '1' en la Fig. 6). Esto nos abre una nueva ventana, (remarcada en azul en la figura anterior), que presenta diversos parámetros (media, desviación estándar, mínimos, máximos, ..) para cada columna de la tabla. Seguimos interesados en los campos relativos a los movimientos propios: pmra, pmdec y sus respectivos errores empíricos. Nótese que esta información está referida al subset 'M67v1' (indicado por el '2' en la imagen).

Nuestro objetivo aquí es añadir un nuevo filtro a los objetos seleccionados en la etapa anterior con la finalidad de acotar en lo posible y de manera razonable los candidatos a ser componentes del cúmulo.

Para ello vamos a tener en cuenta el valor medio de $pmra$ y $pmdec$, así como la dispersión de estos respecto a los valores medios tomando como referencia la *desviación estándar*. También tendremos en cuenta la 'calidad de los datos empíricos' recuperados de Gaia, por lo que los campos $pmra_error$ y $pmdec_error$ serán significativos, pero en este caso no los que presenta la información estadística, sino los correspondientes a cada objeto de 'M67v1'. Así pues construiremos nuestro nuevo filtro en base a las siguientes condiciones:

1. descartar aquellas estrellas que presenten valores en $pmra$ y $pmdec$ que estén fuera del intervalo ' $media(pmra) \pm desviación\ estándar(pmra)$ '. Lo mismo haremos con $pmdec$.
2. descartar las estrellas cuyo error relativo en $pmra$ y $pmdec$ sea superior al 10%. Esto es:
 $pmra_error / pmra > 0.1$ y $pmdec_error / pmdec > 0.1$.

Evidentemente estos criterios pueden ser modificados según se estime oportuno. En el caso práctico que aquí estamos desarrollando serán los utilizados a modo de ejemplo.

Ya estamos listos para construir nuestro nuevo filtro. Ver indicaciones a partir de la Fig. 7.

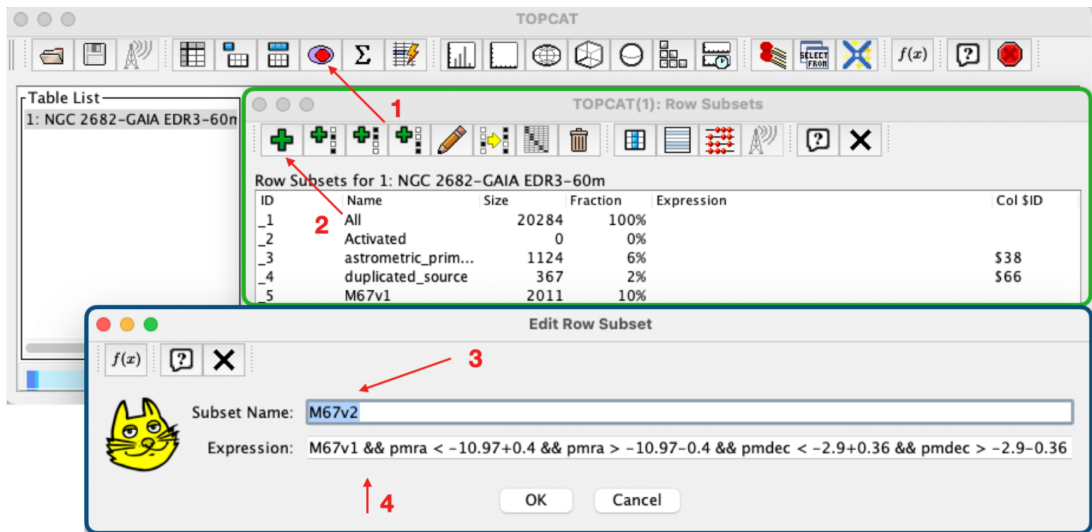


Figura 7. Detalle de un nuevo filtro para 'M67v1'. Remarcados en la figura los sucesivos pasos para hacerlo efectivo. Las indicaciones se detallan en el texto.

Para definir un filtro y generar un nuevo subset a partir de 'M67v1' utilizaremos la función '*display row subsets*', marcada con '1' en la Fig. 7. Se abrirá una ventana, (remarcada en verde en la figura). Ahí, paso '2', seleccionamos '*definir nuevo subset usando una expresión algebraica*' que nos abrirá la ventana remarcada en azul. Damos nombre al nuevo subconjunto, (paso '3', seguimos con el criterio sencillo de añadir un número secuencial a nuestro juego de subsets), y finalmente en la 'caja' destinada a la 'expresión' escribimos nuestro conjunto de reglas sin olvidar que queremos aplicarlas al último subset válido que ya tenemos, (en este caso 'M67v1'), y que para no olvidarlo lo escribiremos en primer lugar:

Expression:

M67v1 &&

$$\begin{aligned} pmra < -10.97 + 0.4 \quad \&\& \quad pmra > -10.97 - 0.4 \quad \&\& \\ pmdec < -2.9 + 0.36 \quad \&\& \quad pmdec > -2.9 - 0.36 \quad \&\& \\ pmra_error/pmra < 0.1 \quad \&\& \quad pmdec_error/pmdec < 0.1 \end{aligned}$$

Las cuatro líneas de la expresión anterior deben escribirse seguidas en una misma línea continua dentro de la caja 'expression'. El doble '&' (&&) es la forma de indicar que todas las condiciones deben cumplirse, (operador lógico 'Y').

También se pueden aplicar por separado de manera sucesiva, pero en este caso debe ponerse especial atención en el *subset* sobre el que se van ejecutando, cuyo nombre deberá estar siempre presente en cada una de las expresiones. Los valores (-10.97, 0.4, -2.9, 0.36) insertados en la expresión del filtro anterior corresponden al valor medio y desviación estándar de cada variable (ver Fig. 6).

Ejecutado el filtro, el nuevo subconjunto 'M67v2' tiene ahora 1.130 estrellas y ya estamos preparados para tener en cuenta la segunda de las propiedades enumeradas en el apartado 2. sobre los OCs: sus componentes estarán a una misma distancia respecto al origen en el sistema de referencia desde el que se observan, dentro de un intervalo razonable propio del carácter extenso de un cúmulo abierto.

3.3. Análisis de distancias. El paralaje

En la información que tenemos de NGC 2682, recuperada de Gaia, la distancia está disponible en el campo 'parallax', esto es en forma de paralaje y unidades 'mas' (milisegundos de arco). Es fácil trasladar este dato a *parsec* (pc), o bien a *años-luz*, (' $distancia(pc) = 1000/parallax(mas)$ ' y ' $distancia(años - luz) = distancia(pc) \cdot 3.26$ '). Pero no es necesario, seguiremos trabajando con el valor del *paralaje*.

Vamos a TOPCAT de nuevo (ver Fig. 8). Ahora seleccionamos la función de '*representar un histograma*' para el valor de paralajes, (en la ventana principal de TOPCAT, paso '1' de la figura). Se abre una nueva ventana, remarcada en la imagen. Sólo debemos seleccionar el campo 'parallax' para el eje X, (paso '2'). En la pestaña 'subsets' se pueden activar los subconjuntos representados, (por ahora es interesante marcar 'all', 'M67v1' y 'M67v2'), y tendremos la representación mostrada en la parte izquierda de la figura (véase leyenda, marcada con 'a'). Se puede ajustar la escala de la representación pulsando la 'cruz' en la parte superior (cuarto icono por la izquierda).

Podemos jugar un poco con la escala de la representación en ambos ejes para tener una visión más detallada de esta distribución (ver Fig. 9), donde se aprecia mejor cómo impacta el subconjunto 'M67v2' sobre el fondo de estrellas. En otros casos, (OCs más cercanos, por ejemplo), el histograma mostraría un 'pico' sensiblemente desplazado sobre la distribución del total. Evidentemente, no son esperables paralajes negativos. Los que aparecen aquí responden a datos empíricos de mala calidad. No nos afectan y los ignoramos.

En la parte central de la Fig. 8 se muestra sólo la distribución para 'M67v2'. Como se hizo en el apartado anterior con las velocidades, vamos a intentar estrechar la dispersión en distancias acotando valores y filtraremos igualmente estrellas con errores relativos mayores a un 10 % en el valor del paralaje. Para ello recurrimos de nuevo a los datos estadísticos de 'M67v2' y tomamos nota de la 'media' y la 'desviación estándar' del campo 'parallax'. Seguiremos el mismo criterio que con los movimientos propios.

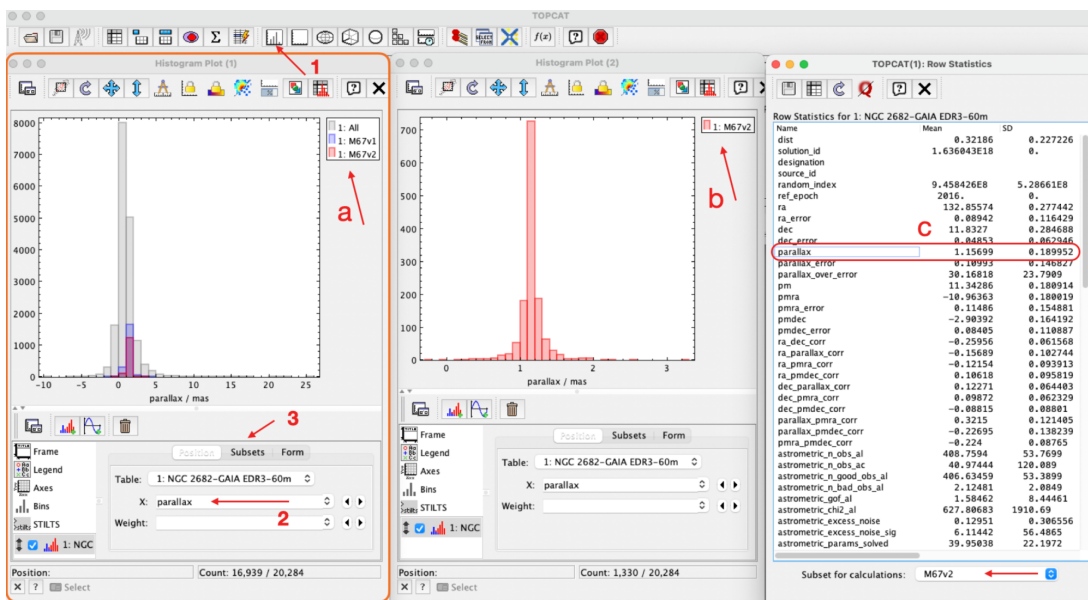


Figura 8. histograma que representa la distribución del campo parallax. En la figura se presentan tres ventanas diferentes (izquierda, centro y derecha) cuyos detalles se comentan en el texto del artículo.

La expresión para el nuevo filtro sería como se muestra a continuación y lo aplicamos para crear un nuevo *subset* 'M67v3' de igual forma que se hizo antes con el resultado que se muestra en la Fig. 10.

M67v2 &&

$$parallax < 1.16 + 0.19 \quad \&\& \quad parallax > 1.16 - 0.19 \quad \&\& \\ parallax_error / parallax < 0.1$$

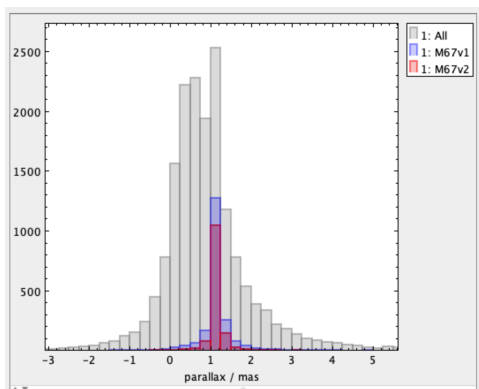


Figura 9. Detalle de la Fig. 8 (izquierda) con escalas en los ejes ajustadas para una mayor resolución de la región de interés.

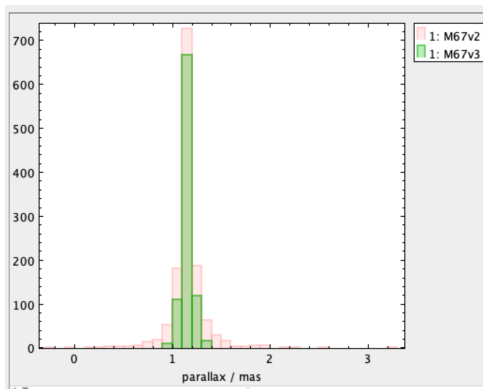


Figura 10. Histograma de parallax para 'M67v3' sobre el fondo de 'M67v2'. En la Fig. 11 se presenta con mayor detalle.

El resultado de los dos filtros aplicados sobre la selección inicial, hecha a *'mano alzada'* a partir del mapa de velocidades del conjunto total, nos deja un *dataset* con **920 estrellas** que damos por bueno en el ejemplo desarrollado aquí. Si comparamos el número de componentes obtenido para NGC 2682 después de este simple análisis con el ofrecido en la *WiKi* [10] veremos que resulta bastante aproximado. Comparado con los datos del catálogo CDS-VizieR [11], el resultado excede sensiblemente a los registrados allí, pero hay que tener en cuenta que hemos ampliado de forma notoria el radio del campo de cielo y también, y esto es más importante, que los datos de Gaia son más completos que los recogidos en *surveys* anteriores.

Si analizamos el resumen estadístico de los datos de nuestra última tabla, *'M67v3'*, y en concreto los relativos a la distancia, *'parallax'*, algo que queda bien representado en el histograma de paralaje (Fig. 11), tendremos una estimación de su distancia relativa a nuestra posición.

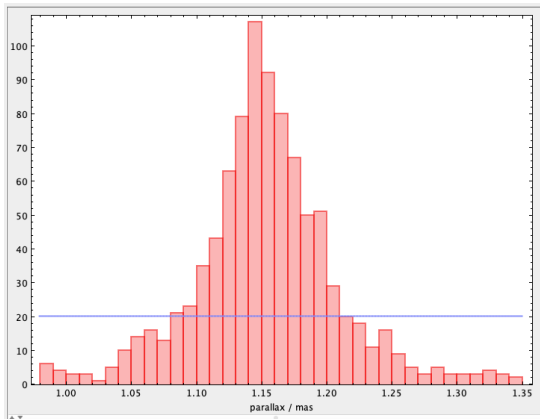


Figura 11. *histograma ampliado de parallax para 'M67v3'. La línea de corte sobre el valor '20' en el número de estrellas se corresponde muy aproximadamente con una nueva estimación del tipo $media(parallax) \pm sd$ (desviación estándar), que en términos numéricos es 1.153 ± 0.055 .*

Asumiendo una distancia media al cúmulo de 1.153 (*mas*) con un intervalo de confianza del $68\% = 0.055$ (*mas*)³, tendremos una distancia de $867.3(pc) \pm 42(pc)$, (recordar que $distancia(pc) = (1/parallax(mas)) \cdot 1000$), o bien en *años-luz* $= 2.827 \pm 137$ que sería un rango razonable de amplitud en su profundidad de distancia.

Todavía no podemos estimar datos como la masa total del cúmulo, luminosidad, composición química, edad, etc. Para terminar de caracterizarlo, una vez que asumimos cuáles son las estrellas componentes, deberíamos recabar datos de cada una de ellas y aplicar algunos modelos *ad hoc* para este cometido. Intentaremos alguna aproximación en la tercera entrega de esta serie.

Lo que si podemos hacer es transferir *'M67v3'* a Aladin con objeto de reproducir la Fig. 1, tal como ya se comentó en la parte inicial de este artículo. Pero antes de terminar, aún podemos hacer algo más ...

4. El diagrama color-magnitud

Sin abandonar TOPCAT vamos a representar el diagrama color-magnitud de *'M67v3'*, similar a un diagrama *Hertzsprung-Russell*, pero en vez de flujos y temperaturas utilizaremos el *índice de color* (*azul-rojo*) y la *magnitud visual* (*gaia*), aprovechando que en la información descargada de Gaia tenemos también estos datos fotométricos debidamente calibrados. Para ello volvemos a representar un mapa con estos dos parámetros en los respectivos ejes X e Y. Ver Fig. 12.

De nuevo seleccionamos la función de *'display plane window'*, (paso '1' en la Fig. 12), que nos abrirá una nueva ventana en la que asignaremos el campo *'bp-rp'* al eje X y *'phot-g-mean-mag'* en el eje Y,

³En una distribución normal el rango *'media ± σ'* incluye el 68% de la muestra.

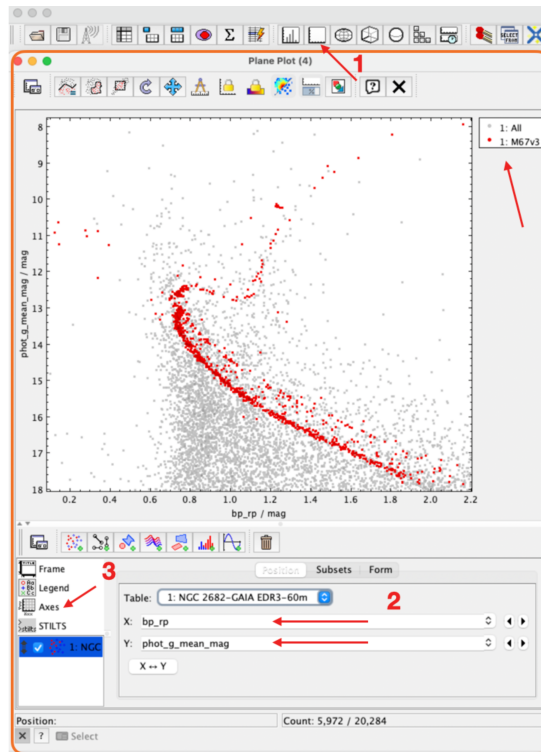


Figura 12. Diagrama color-magnitud para 'M67v3' con el fondo del conjunto inicial descargado para el área de cielo de 1º centrada en las coordenadas de NGC 2682. Los detalles en el texto del artículo.

(paso '2'), (esto es 'magnitud en el azul - magnitud en el rojo = índice de color (b-r) y magnitud visual respectivamente).

Será conveniente invertir el eje Y en esta representación para ver el diagrama en su forma más familiar. Para ello, en la parte inferior izquierda, se selecciona la opción 'ejes' y se procede sobre el eje Y, (paso '3'). Como ya sabemos, en la pestaña 'subsets' podemos activar los subconjuntos a visualizar y asignarles los colores deseados. En la Fig. 12 se presenta el subset 'M67v3', esto es, la última versión de este breve análisis después de los filtros aplicados y, de fondo, el conjunto total inicial. Si nos fijamos en el pie del gráfico se muestra un contador que indica *Count: 5,972 / 20,284* y que significa que, de los 20284 objetos recuperados de Gaia, sólo 5972 se han representado; el resto carecen de alguno de los datos dibujados o de ambos. De las 920 estrellas de 'M67v3', 919 están incluidas en el gráfico.

Como puede apreciarse en el gráfico el conjunto de estrellas (en rojo), que finalmente hemos atribuido a NGC 2682 como componentes del mismo, muestran un perfil bastante nítido en el que se aprecia como la mayoría de las estrellas están en la *secuencia principal* y también se advierte el punto de inflexión en el que ya algunas de ellas inician su recorrido hacia la región de gigantes y super-gigantes rojas. Un número muy pequeño se observa en la zona de las estrellas más brillantes (azules y/o blancas). De aquí puede deducirse que éste, además de ser un cúmulo bastante poblado, no es un OC especialmente joven. Aunque para intentar la aproximación a una determinación más precisa de su edad y otras características del mismo tendremos que esperar a la tercera entrega de esta serie de artículos, en la que haremos uso de la herramienta VO 'VOSA' [4].

5. Otros aspectos de interés

Antes de terminar con TOPCAT hay que advertir que una de las funcionalidades de esta herramienta, (segundo icono por la izquierda en la ventana principal), permite guardar localmente bien una o todas las tablas utilizadas y, así mismo, la sesión completa, con los filtros y procesos desarrollados. Hay varios formatos disponibles para salvar estos objetos.

Igualmente podemos tener una representación 3D del cúmulo tanto del mapa de posiciones espaciales (ejes: $X = RA$, $Y = Dec$ y $Z = parallax$ en coordenadas cartesianas, Fig. 13), así como una imagen del *chorro* dinámico que presenta frente a su entorno más próximo a partir de un mapa 3D sobre los movimientos propios en coordenadas cilíndricas, Fig. 14, (ejes: $X = pmra$, $Y = pmdec$ y $Z = parallax$).

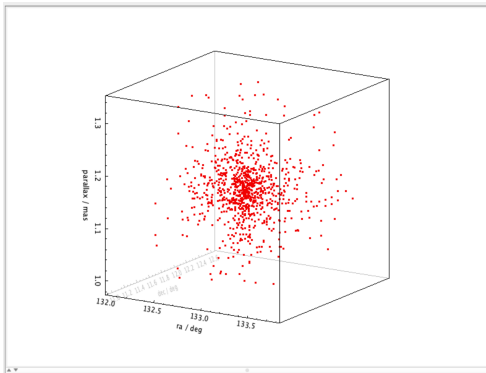


Figura 13. Mapa 3D de NGC 2682 en coordenadas cartesianas, (RA , Dec), en el eje Z se representa la distancia. Esta funcionalidad en TOPCAT permite realizar movimientos de rotación en los tres ejes para una visualización más completa.

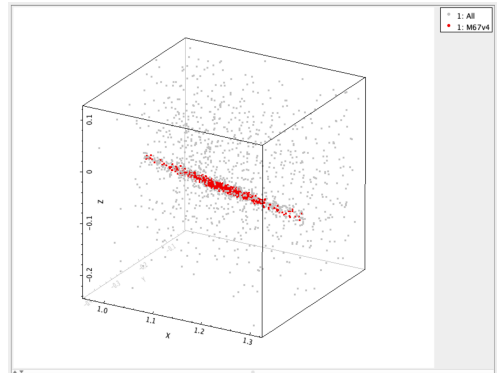


Figura 14. Mapa 3D en coordenadas cilíndricas del espacio de configuración de velocidades, (en el eje Z se representa también la distancia). Esta representación da idea de la dinámica propia del cúmulo, en forma de 'chorro', en relación a su entorno de fondo de cielo.

Puesto que tenemos una estimación del movimiento propio del cúmulo podríamos aventurar también una evolución temporal de su posición relativa en la galaxia, tanto hacia el futuro como al pasado, por ejemplo en millones de años, (10^6 , 10^7 , ...). Para esto sería conveniente utilizar coordenadas galácticas y hacer las correspondientes transformaciones de los valores con los que hemos trabajado (ra , dec , $pmra$ y $pmdec$) al nuevo sistema de referencia. (TOPCAT permite añadir campos nuevos a cualquiera de las tablas en curso definiendo al tiempo su contenido, de esta manera sin demasiadas complicaciones se pueden realizar simulaciones como las comentadas aquí). Un ejemplo al respecto podemos verlo en *Cameren Swiggum, et al.* [14], artículo en el que se detallan 3 superestructuras, o *familias* de cúmulos *cercanos*, generadas a partir de *macro-nubes* de gas en una simulación hacia atrás en el tiempo [$30 \cdot 10^6$ años] y circunscrita a un espacio de 6500 años-luz en torno al Sol [15].

En el próximo artículo haremos uso de *Clusterix 2.0* [3], otra herramienta del Observatorio Virtual que facilita sensiblemente el trabajo que hemos hecho hasta aquí, al menos en su primera parte. Como hemos comentado en algún momento, hay cierto grado de *artesanía* en todo esto. En realidad un procesado de este tipo debemos definirlo como *supervisado* y *parametrizado* en tanto que exige en todo momento la actuación del analista definiendo criterios y siguiendo paso a paso su desarrollo. *Clusterix* se define a sí mismo como herramienta *no supervisada* y *no parametrizada*, aunque ya veremos que tiene otros condicionantes . . .

Agradecimientos

TOPCAT -

This research has made use of "TOPCAT VO"

Author: Mark Taylor, Astrophysics Group, Physics Department, University of Bristol

Email: m.b.taylor@bristol.ac.uk

TOPCAT WWW page: <http://www.starlink.ac.uk/topcat/>

<https://ui.adsabs.harvard.edu/abs/2005ASPC..347...29T/abstract>

Aladin -

This research has made use of "Aladin sky atlas" developed at CDS, Strasbourg Observatory, France

<https://ui.adsabs.harvard.edu/abs/2000A%26AS..143...33B/abstract>

<https://ui.adsabs.harvard.edu/abs/2014ASPC..485..277B/abstract>

<https://ui.adsabs.harvard.edu/abs/2022ASPC..532....7B/abstract>

Spanish Virtual Observatory-

This research has made use of the Spanish Virtual Observatory <https://svo.cab.inta-csic.es> project funded by MCIN/AEI/10.13039/501100011033/ through grant PID2020-112949GB-I00.

References

- [1] TOPCAT: Tool for OPerations on Catalogues And Tables - <https://www.star.bris.ac.uk/~mbt/topcat/>
- [2] Aladin Sky Atlas - <https://aladin.cds.unistra.fr>
- [3] Clusterix 2.0 - <http://clusterix.cab.inta-csic.es/clusterix/>
- [4] VO SED Analyzer - <http://svo2.cab.inta-csic.es/theory/vosa/index.php>
- [5] Clarke, C.J. et al - "The Formation of Stellar Clusters" - <https://ui.adsabs.harvard.edu/abs/2000prpl.conf..151C/abstract>
- [6] Portegies Zwart et al. - "Young Massive Star Clusters" - <https://ui.adsabs.harvard.edu/abs/2010ARA%26A..48..431P/abstract>
- [7] Cantat-Gaudin, T. et al. - "A Gaia DR2 view of the open cluster population in the Milky Way" - https://www.aanda.org/articles/aa/full_html/2018/10/aa33476-18/aa33476-18.html
- [8] Misión Gaia (ESA) - <https://sci.esa.int/web/gaia/>
- [9] Alfred Castro-Ginard et al. - "Hunting for open clusters in gaia dr2: 582 new open clusters in the galactic disc" - Astronomy & Astrophysics, 635:A45, 2020.
- [10] Wikipedia - https://es.wikipedia.org/wiki/Cúmulo_abierto_M67
- [11] Open Clusters - CDS VizieR Catalogue - <http://vizier.cds.unistra.fr/viz-bin/VizieR-3?-source=B/ocl/clusters>
- [12] Spanish Virtual Observatory <https://svo.cab.inta-csic.es>. Escuelas y tutoriales: <https://svo.cab.inta-csic.es/docs/index.php?pagename=Meetings>
- [13] Francisco Jiménez-Esteban & Belén López (SVO) - "Determination of the distance to Pleiades using TOPCAT" - <https://zenodo.org/records/10720670>
- [14] Cameren Swiggum et a. - "Most nearby young star clusters formed in three massive complexes" - <https://arxiv.org/abs/2406.06510>
- [15] YouTube @Cameren Swiggum Astro - <https://www.youtube.com/watch?v=4iBy5hz6eNE>

SECTION: PROGRAMACIÓN DE DISPOSITIVOS ASTRONÓMICOS

Diseño, construcción y programación de dispositivos astronómicos: una introducción

Sergio Alonso¹ and Javier Flores²

¹Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Granada, Sociedad Astronómica Granadina, España.

E-mail: zerjioi@ugr.es.

²Observatorio Astronómico de Calar Alto, Sociedad Astronómica Granadina, España. E-mail: jflores@caha.es.

Palabras Clave: dispositivos astronómicos, *astronomical devices*, *hardware*, *software*, controlador, *driver*, INDI, INDIGO, panel de *flats*, *flats panel*

© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)

Resumen

En este primer artículo de la serie dedicada al diseño, construcción y programación de dispositivos astronómicos hacemos una introducción a como los sistemas operativos actuales son capaces de gestionar los dispositivos que usualmente conectamos a nuestros ordenadores. En el caso particular de la astronomía, por la especificidad de los dispositivos que manejamos (que pueden no ser de interés para el usuario general de la informática) además existen varias plataformas de más alto nivel (ASCOM, INDI, INDIGO) que permiten tanto el desarrollo de nuevo instrumental como el control del mismo, incluso de manera distribuida. Hacemos un repaso de las características principales de dos de estas plataformas (INDI e INDIGO) y presentaremos de manera breve el dispositivo (un panel de *flats/dark*) que usaremos a modo de ejemplo en los siguientes artículos de la serie.

Abstract

In this first article of the series dedicated to the design, construction and programming of astronomical devices we make an introduction to how current operating systems are able to manage the devices that we usually connect to our computers. In the particular case of astronomy, due to the specificity of the devices we handle (which may not be of interest to the general computer user) there are also several higher level platforms (ASCOM, INDI, INDIGO) that allow both the development of new instruments and their control, even in a distributed environment. We review the main features of two of these platforms (INDI and INDIGO) and briefly introduce the device (a *flats/dark* panel) that we will use as an example in the following articles of the series.

Introducción a la Sección

Presentamos una serie de cuatro artículos que pretenden facilitar al lector la tarea del diseño, construcción y programación de un dispositivo astronómico que pueda resultarle de utilidad en sus observaciones. En el desarrollo de la serie de artículos se utilizarán diversas tecnologías libres (y en muchos casos gratuitas o de bajo coste) y lenguajes de programación (como Python, Arduino, INDIGO, diseño paramétrico...) que permitan abordar las distintas tareas necesarias para conseguir un dispositivo controlable remotamente y completamente funcional. Los cuatro artículos se organizarán de la siguiente manera:

- Este primer artículo expone algunos de los conceptos necesarios para poder desarrollar nuestro dispositivo y todo el software que permitirá su control usando estándares y tecnologías actuales.

Además se presentará de manera sucinta un dispositivo astronómico que servirá a modo de ejemplo para el desarrollo durante toda la serie.

- El segundo artículo se centrará en el diseño del *hardware*, *firmware* y el protocolo de comunicación con el dispositivo. Se hará uso del software FreeCAD para el diseño de las piezas físicas del dispositivo (que podrán ser impresas en 3D), así como la plataforma Arduino (junto a su lenguaje de programación) para el control de los componentes electrónicos necesarios para el dispositivo.
- El tercer artículo abordará la programación de un *driver* INDIGO que permita el control del dispositivo desarrollado haciendo efectiva la comunicación entre el dispositivo basado en Arduino y el ordenador al que se conectará. Una vez finalizado el *driver* del dispositivo, estaremos en condiciones de poder controlarlo mediante cualquier *software* que implemente el protocolo INDI, como pueden ser KStars, Ekos, AstroDMx Capture, Cartes du Ciel, PixInsight, Stellarium...
- El cuarto y último artículo de la serie planteará la programación de un programa cliente que haga uso de nuestro dispositivo por sí los programas de control de dispositivos astronómicos habituales no nos permitieran realizar algún tipo de tarea particular con el mismo (o no lo hagan de manera satisfactoria).

1. Introducción: *hardware*, abstracciones y dispositivos astronómicos

Desde el mismo comienzo de la informática, cuando los primeros diseños sobre hipotéticas máquinas electrónicas capaces de procesar datos y resolver problemas con los mismos empezaban a aparecer, se hacía patente que dichos proto-ordenadores necesitarían de dispositivos de entrada y salida, es decir, mecanismos que les permitieran introducir y extraer la información (los datos y los programas) de los mismos.

Por ejemplo, el mismo John von Neumann en el primer borrador de un informe sobre el EDVAC[1] (uno de los primeros ordenadores electrónicos) donde se describen las bases sobre las que se construirían los primeros ordenadores electrónicos conceptualmente similares a los que manejamos actualmente (arquitectura von Neumann[2]), se mencionan específicamente los dispositivos de entrada y salida.

Evidentemente los dispositivos de entrada y salida han evolucionado exponencialmente junto con el resto de componentes de los ordenadores: desde interruptores básicos y tarjetas perforadas como mecanismos de entrada y luces e impresoras básicas como mecanismos de salida, hasta sensores de alta sensibilidad y resolución (por ejemplo, las cámaras CCD) como dispositivos de entrada y pantallas de alta resolución o incluso impresoras 3D como dispositivos de salida actuales.

De hecho, la amplia variedad de dispositivos de entrada y salida (de cualquier fabricante o incluso de fabricación artesanal) ha provocado que la gestión de todos los dispositivos que podamos llegar a conectar en nuestros ordenadores no sea una tarea trivial. Cuando los primeros ordenadores comienzan a ser accesibles por el público general, si comprábamos algún dispositivo nuevo (pongamos, por ejemplo, una nueva impresora que acabara de salir al mercado) era posible que los programas de ordenador (un *software* de edición de textos, por continuar con el ejemplo) que utilizáramos no fueran capaces de manejar dicho nuevo *hardware* por la sencilla razón de que dichos programas habrían sido creados antes de que el dispositivo en cuestión existiera.

Para solucionar este tipo de casuísticas es por lo que los sistemas operativos modernos[3] tienen la gestión del hardware de los distintos dispositivos que se pueden conectar a nuestros ordenadores como una de sus funciones principales. Esta labor la llevan a cabo mediante la inclusión de la denominada Capa de Abstracción Hardware (*Hardware Abstraction Layer, H.A.L.*)[4]. Dicha capa consiste en un mecanismo que permite definir abstracciones de *hardware* de tal manera que los programadores de aplicaciones ya no interactúan directamente con los dispositivos conectados al ordenador, sino que programarán contra una interfaz *software* que hará de "puente" entre el dispositivo físico y la aplicación propiamente dicha.

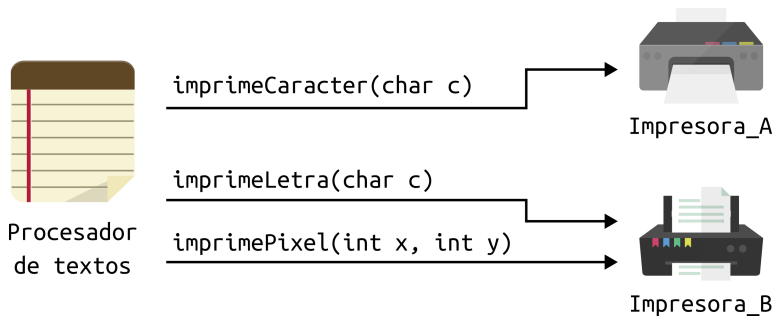


Figura 1. Esquema simple donde un programa procesador de textos interactúa con dos impresoras: debe conocer el funcionamiento de cada una de ellas. Iconos por [SVG Repo](#).

Por clarificar la utilidad de utilizar un mecanismo de abstracción de *hardware*, continuaremos con el ejemplo de impresoras mencionado anteriormente: Suponemos que existe una impresora simple (Impresora_A) que podemos hacer funcionar invocando una función llamada `imprimeCaracter(char c)`. Dicha función simplemente imprimirá el carácter `c` que le hayamos pasado como parámetro. Supongamos que aparece en el mercado una nueva impresora más avanzada (Impresora_B). Al ser de un fabricante distinto, incluye las siguientes funciones que podemos invocar desde nuestro programa:

- `imprimeLetra(char c)`, completamente análoga a la función de la otra impresora (pero distinta, puesto que cada fabricante construye sus dispositivos siguiendo patrones distintos).
- `imprimePixel(int x, int y)`, que imprimiría un punto negro en unas coordenadas `x` e `y` del papel.

Para que un programa de ordenador pudiera manejar ambas impresoras, el programador de la aplicación tiene que conocer el funcionamiento de ambas impresoras (es decir, conocer qué funciones puede usar con la impresora concreta que tiene que utilizar). Esto hace muy compleja la labor del programador, puesto que es imposible estar al tanto de todas las impresoras existentes y su funcionamiento (Fig. 1). En este momento aparece el mecanismo de abstracción de *hardware*: El diseñador del sistema operativo piensa en la problemática que existe con la cantidad de impresoras distintas que existen en el mercado y define una abstracción llamada `Impresora`. Dicha abstracción no es una impresora concreta, ni se refiere a una marca o modelo concreto. Simplemente es la definición genérica de las instrucciones que debe llevar a cabo un dispositivo para que el sistema operativo la considere una impresora. En nuestro caso dicha abstracción podría consistir en la definición de las siguientes funciones:

- `imprimeCadena(String s)`: Esta función debería imprimir una secuencia de caracteres en la impresora.
- `puedeImprimirGraficos()`: `boolean`: Esta función devolvería `true` si la impresora es capaz de imprimir gráficos y `false` en caso contrario.
- `imprimeGrafico(boolean[][] pixeles)`: esta función imprimiría una gráfico a partir de la información en el array de `pixeles` que se le pasa por parámetro.

Es importante recalcar que la definición abstracta de `Impresora` no ha implementado el como se imprime con ninguna impresora particular. De hecho, solo con esa definición el sistema operativo (ni ninguna aplicación instalada en el mismo) aún no es capaz de imprimir en ninguna impresora concreta puesto que el diseñador del sistema ni de las aplicaciones no tienen por qué conocer como funcionan las impresoras fabricadas por atrás empresas. De hecho, son dichos fabricantes de *hardware* los que

tienen que proporcionar los controladores (comúnmente denominados *drivers*) que serán los encargados de que sus dispositivos sean utilizables a través del sistema operativo. En ese sentido, un *driver* no es ni más ni menos que un "pequeño" programa de ordenador que hace de intermediario ("traduce") entre las instrucciones abstractas definidas en la abstracción *Impresora* a las instrucciones concretas que necesita el dispositivo concreto. Por tanto, el fabricante de la *Impresora_A* deberá construir un *driver* cuyas funciones serían:

- `imprimeCadena(String s)`: Esta función hará uso de la función `imprimeCaracter(char c)` de manera iterativa para cada uno de los caracteres de la cadena `s` para imprimir la cadena completa.
- `puedeImprimirGraficos()`: `boolean`: En este caso devolverá `false`, puesto que la impresora no es capaz de imprimir gráficos.
- `imprimeGrafico(boolean[][] pixeles)`: esta función no haría nada, puesto que la impresora no es capaz de imprimir gráficos.

En el caso del segundo fabricante, deberá proporcionar un driver para la *Impresora_B* de la siguiente manera (usando las funciones propias de dicha impresora):

- `imprimeCadena(String s)`: Esta función hará uso de la función `imprimeLetra(char c)` de manera iterativa para cada uno de los caracteres de la cadena `s` para imprimir la cadena completa.
- `puedeImprimirGraficos()`: `boolean`: En este caso devolverá `true`, puesto que la impresora si puede imprimir gráficos.
- `imprimeGrafico(boolean[][] pixeles)`: Esta función imprimirá la imagen llamando iterativamente a la función `imprimePixel(int x, int y)` de la impresora para cada pixel definido en `pixeles`.

De esta manera, el programador de una aplicación (por ejemplo un procesador de textos) ya no tiene que conocer como funciona cada una de las impresoras para usarlas. Solo debe conocer la abstracción *Impresora* y manejar las tres funciones que se definen en dicha interfaz, puesto que el sistema operativo se encargará de pasar las llamadas a las funciones de *Impresora* al driver de la impresora que corresponda. Y lo que es más importante, cuando aparezca una nueva impresora en el mercado, podrá ser usada por los programas existentes (sin tener que modificarlos) puesto que dicha impresora implementará exactamente las mismas funciones definidas en la abstracción *Impresora*. Este esquema, más complejo pero mucho más flexible, puede verse esquematizado en la Fig. 2.

Es importante enfatizar que es obligación de los fabricantes de *hardware* proveer los *drivers* para su dispositivo para cada sistema operativo en los que quieran que funcione su instrumento. En ningún caso es tarea del sistema operativo o de una tercera parte el programar dicho *driver* aunque en ocasiones los propios sistemas operativos o comunidades de programación ajenas a los fabricantes hayan desarrollado *drivers* para muchos dispositivos populares, en algunos casos llevando a cabo una labor de ingeniería inversa importante.

La mayoría de los dispositivos actuales que conectamos a nuestros ordenadores tienen algún tipo de interfaz de comunicación para conectarlo. En la actualidad se ha impuesto el interfaz USB (Bus Serie Universal), aunque siguen existiendo otros puertos en uso, como el puerto serie, paralelo, *firewire*, RJ45 (puerto de red), etc. A través de este tipo de interfaces los dispositivos pueden recibir comandos y responder a los mismos usando algún tipo de protocolo de comunicaciones usualmente definido por el fabricante.

En el caso particular de la astronomía, donde los usuarios manejan una gran cantidad de dispositivos muy específicos y no de uso común fuera de este ámbito (como por ejemplo ruedas porta-filtros, enfocadores, monturas, calentadores...), no cabe esperar que los diseñadores de los sistemas operativos

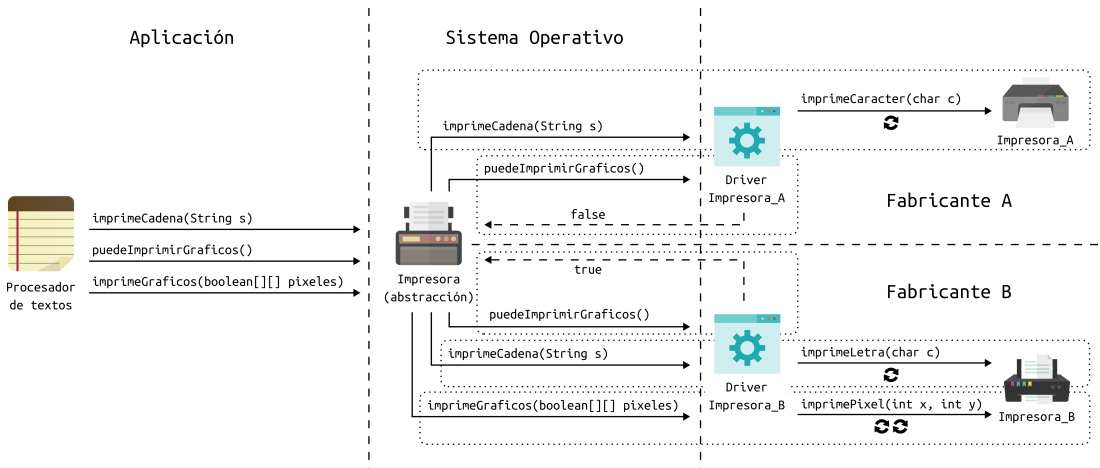


Figura 2. Esquema de aplicación de una H.A.L.: el sistema operativo provee una abstracción impresora y la aplicación por tanto no tiene que conocer los detalles de implementación de cada impresora, que es tarea específica de los drivers. Iconos por SVG Repo.

desarrollen abstracciones específicas para cada uno de estas tipologías de dispositivos. Es por ello que han aparecido varias plataformas (la mayoría de ellas libres[10]) que, a un nivel más alto (por encima del sistema operativo), ofrecen características similares a las de una H.A.L. pero centradas específicamente en dispositivos de interés para la comunidad astronómica. Una de esas plataformas es ASCOM [5] que, basada principalmente en el ecosistema Windows, lleva desde 1998 creando una serie de interfaces y abstracciones para poder controlar dispositivos astronómicos. En la actualidad sigue siendo muy utilizado y la mayoría de fabricantes de *hardware* astronómico suelen proporcionar un *driver* ASCOM para sus dispositivos. Sin embargo, la plataforma ASCOM ha estado siempre muy centrada en el control de dispositivos conectados directamente al ordenador donde el usuario ejecuta su *software* de control. Sin embargo, con la evolución rapidísima de las redes de comunicaciones y las facilidades y velocidad que estas nos ofrecen, la astronomía moderna se está transformando y permitiendo de manera cada vez más sencilla controlar instrumental astronómico en remoto (por ejemplo en zonas geográficas menos contaminadas) e incluso de manera distribuida, es decir, manejando de manera simultánea equipamiento que puede encontrarse en diversas localizaciones (para sincronizar observatorios, o simplemente para manejar gran cantidad de instrumental que no podría funcionar conectado a un único ordenador.

Para poder llevar a cabo este tipo de control remoto de instrumental han aparecido varias iniciativas que pretenden facilitar dicha tarea. INDI e INDIGO son algunas de ellas, que describiremos en las siguientes secciones, puesto que serán sobre las que en los siguientes artículos de la serie se desarrollará el código de control de nuestro dispositivo de ejemplo. Pero antes de presentar sendas iniciativas, es de justicia señalar que en el año 2018 los desarrolladores de la plataforma ASCOM, viendo el interés creciente en el control remoto de observatorios (que usualmente los usuarios de ASCOM realizaban mediante algún tipo de aplicación de control remoto de escritorio) empezaron a desarrollar ASCOM Alpaca[6], una iniciativa que permitiría controlar dispositivos remotamente (usando un protocolo similar a REST que usa `http` y `json` para comunicarse por la red) e incluso bajo sistemas operativos diferentes a Windows. Sin embargo, dicha iniciativa, pese a que es funcional, todavía no está tan extendida como la plataforma ASCOM original.

2. INDI: una aproximación distribuida al control de dispositivos astronómicos

INDI[7] (*Instrument Neutral Distributed Interface*) es un sistema de control distribuido de instrumental, pensado originalmente para astronomía, pero que en realidad puede usarse para cualquier tipo de dispositivos. Nace en el año 2003 con la publicación de la primera versión del protocolo INDI[8], que está basado en XML y, por tanto, es un protocolo fácil de leer y escribir por cualquier lenguaje de programación moderno (existen numerosas bibliotecas que permiten leer y escribir dicho protocolo).

Además del protocolo en sí, con el tiempo se han ido desarrollando diversos paquetes de software que implementan dicho protocolo, como pueden ser diversos clientes (Ekos, Kstars, Cartes du Ciel, Stellarium...) pero, sobretodo la biblioteca INDILib[9] que es una implementación de referencia, basada en el lenguaje de programación C y disponible para sistemas operativos de tipo UNIX (como por ejemplo GNU/Linux) que no solo implementa el protocolo de comunicación INDI, sino que también proporciona algunas herramientas fundamentales como son un Servidor INDI y la mayoría de los *drivers* INDI disponibles para el instrumental astronómico. Es interesante reseñar que existen implementaciones del protocolo INDI que permiten la creación de *drivers*, clientes y servidores en otros lenguajes como Java[11] o Python[12].

En lo que sigue de esta sección haremos una descripción general del sistema INDI, sin prestar especial atención al protocolo de comunicación en sí, puesto que dicho protocolo no es necesario conocerlo para poder usar e incluso programar nuestros propios clientes o *drivers* INDI.

2.1. Arquitectura de la plataforma INDI

INDI nace asumiendo desde el principio que el control de los distintos dispositivos de los que disponemos se hará de manera distribuida. Es decir, los dispositivos que queramos controlar no tienen por qué estar conectados en el propio ordenador desde donde deseamos controlarlos. De hecho, la propia arquitectura permite de manera simple controlar dispositivos que pueden estar conectados a varios ordenadores en diversas localizaciones, siempre y cuando haya un enlace de red a los mismos. Incluso se pueden controlar y monitorizar los distintos dispositivos desde varios ordenadores diferentes.

Para conseguirlo, INDI distingue tres actores software distintos:

- **Ciente INDI:** Es el programa de ordenador que maneja el usuario que quiere controlar los dispositivos. Usualmente tendrá algún tipo de interfaz de usuario (por ejemplo una ventana gráfica o un interfaz en línea de comandos).
- **Driver INDI:** Es un programa de ordenador que se encarga del controlador de un dispositivo concreto. Dicho programa cumple con la función clásica de los *drivers* de dispositivo en los sistemas operativos: usando una serie de abstracciones permite controlar un dispositivo concreto.
- **Servidor INDI:** Es el programa de ordenador que permite la comunicación (hace de intermediario) entre los clientes y los *drivers*. El servidor estará instalado en el ordenador (o los ordenadores) al que estén conectados los dispositivos. Además se encarga de ejecutar los *drivers* y es el que permite múltiples conexiones de clientes para controlar los dispositivos.

En la Fig. 3 se muestra un esquema simple de conexión entre clientes, servidores y *drivers* INDI. En este caso 2 clientes distintos (uno en un ordenador portátil y otro en un dispositivo móvil) quieren supervisar y controlar tres dispositivos astronómicos distintos: Una cámara, una montura y una cúpula. Es interesante resaltar que los clientes INDI pueden tener naturalezas muy distintas: pueden ser una aplicación web, una aplicación "de escritorio" (para casi cualquier sistema operativo), una App de un dispositivo móvil, etc. Por motivos de organización del observatorio a controlar, la cámara está conectada a un ordenador (con un interfaz USB rápido), mientras que la montura y la cúpula estarán conectadas a otro ordenador distinto. En sendos ordenadores se instarán y lanzarán dos instancias del

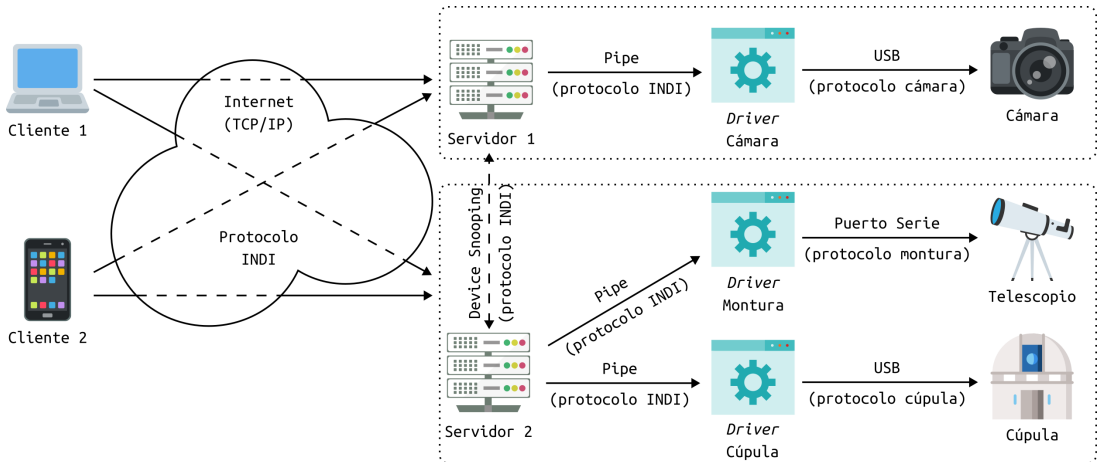


Figura 3. Iconos por SVG Repo.

servidor INDI. Cada una de ellas ejecutará y se comunicará con los *drivers* INDI para los dispositivos conectados. Dicha conexión entre los servidores y los *drivers* INDI se lleva a cabo a través de *pipes* [14] (tuberías) estándar de UNIX. En este ejemplo asumimos que los clientes (tanto el portátil como el dispositivo móvil) no se encuentran ni siquiera en la misma red local que los servidores y que se conectan a los mismos a través de Internet usando *sockets* (TCP/IP) estándar. En la figura también viene indicada una conexión entre los dos servidores INDI que permite el *device snooping*, es decir, que un dispositivo pueda obtener información de otro distinto de manera directa, sin tener que pasar a través de ningún cliente. Por ejemplo, una cámara podría obtener información directamente de la montura para poder rellenar la información de las cabeceras del fichero FITS que va a devolver. Pese a que las conexiones entre los distintos componentes de la arquitectura INDI pueden ser de naturaleza distinta (*sockets*, *pipes*...), la comunicación entre los mismos siempre se hace usando el protocolo INDI, lo que facilita la programación de los distintos elementos.

2.2. Los dispositivos y drivers INDI

Una vez conocida la arquitectura general que propone INDI debemos conocer como INDI crea las abstracciones de hardware para poder construir los drivers de un dispositivo concreto. Para INDI un dispositivo es una colección de lo que llama *propiedades* (*property*). Una propiedad es un atributo del dispositivo. Existen 5 tipos distintos de propiedades que se pueden definir en INDI:

- **Cadenas de texto (*string*):** Representan una propiedad textual. Por ejemplo, puede ser el nombre del dispositivo, la versión del mismo, la ruta del puerto donde debe conectar el ordenador para comunicarse con el dispositivo...
- **Números:** Representan cualquier característica que pueda definirse mediante un número. En principio los números en INDI están definidos en coma flotante, aunque se permite especificar cual es el mínimo, máximo y paso de la propiedad. Por ejemplo, podríamos definir una propiedad numérica que pudiera tomar como valor mínimo -10.5 , máximo 14.7 y el paso entre un valor y otro fuera 0.1 . Dada la importancia del sistema de numeración sexagesimal en astronomía, los números pueden expresarse también en grados (u horas), minutos y segundos.
- **Luces:** Representan características que pueden representarse por cualquiera de las siguientes etiquetas:

- OK: La propiedad está encendida, activada. Por ejemplo, hemos activado un interruptor de un dispositivo. Normalmente se identifica este estado con el color verde ●.
 - BUSY: La propiedad está ocupada. Por ejemplo, está moviéndose un dispositivo y todavía no está listo para funcionar. Normalmente se identifica este estado con el color amarillo ●.
 - ALERT: La propiedad indica que ha ocurrido algo erróneo o algún tipo de problema o inconsistencia. Por ejemplo, se ha activado algún fin de carrera del dispositivo al que no debería haberse llegado. Normalmente se identifica con el color rojo ●.
 - IDLE: Indica un estado "no inicializado" o "inactivo". Normalmente se identifica con el color gris ●.
- **Opciones (*Switch*):** Representan características que solo pueden tomar algunos valores prefijados. Por ejemplo, una propiedad de este tipo podría ser Sensibilidad ISO y que pudiera tener uno de los siguientes valores: 100, 200, 400, 800, H1, H2. Además las propiedades de tipo *switch* tienen una regla asociada que determina cuantos valores pueden o deben seleccionarse de entre sus opciones: *una de muchos* (debe seleccionarse una y solo una de las opciones), *como mucho una* (puede seleccionarse una opción o ninguna) y *cualquiera de muchas* (se puede seleccionar ninguna, una o más de una de las opciones).
 - **Datos binarios (*BLOB*):** Cualquier tipo de dato binario que no pueda codificarse fácilmente con las propiedades anteriores. El ejemplo típico son los datos asociados a los píxeles captados por una cámara (es decir, la imagen en sí captada por la cámara).

Por afinar un poco más, hay que mencionar que en realidad una propiedad es una colección de uno o más valores (*values*) de alguna de las 5 naturalezas presentadas anteriormente: no tiene por qué referirse a un único valor. Esto es especialmente interesante para, por ejemplo, los datos que deben estar agrupados de manera natural. Por ejemplo, para una montura puede existir una propiedad numérica llamada EQUATORIAL_EOD_COORD que representa la ascensión recta (AR, un número) y declinación (dec, otro número) a la que está apuntando en ese momento.

Además, las propiedades pueden definirse como "de solo lectura" (*read only*), es decir, que el cliente no puede pedir que cambien los valores de la misma (típico de propiedades que representan el estado de un sensor), "de lectura y escritura" (*read/write*), que los clientes si pueden pedir su modificación y de "solo escritura" *write only* (algo bastante más inusual). Por último mencionaremos que las propiedades pueden cada una de ellas tener un estado propio ●OK, ●BUSY, ●ALERT o ●IDLE que determina si el valor de la misma es válido, está cambiando, tiene algún problema o aún no está inicializado.

Cuando un cliente INDI conecta con un servidor INDI, este último le manda una lista con todas las propiedades que tienen disponibles cada uno de los *drivers* que el gestiona. De esa manera, totalmente dinámica, el cliente averigua cuales son las capacidades de cada uno de los dispositivos a los que puede conectar. Este dinamismo, en el que a priori el cliente no conoce cuales son las características de los dispositivos que maneja permite el desarrollo de clientes que van a ser capaces de interactuar perfectamente con futuros dispositivos que aún pueden no haberse desarrollado sin tener que actualizarse explícitamente.

Los clientes pueden pedir a los dispositivos que cambien el valor de sus propiedades, lo cual puede desencadenar que el dispositivo ejecute alguna acción concreta. Por ejemplo, si el dispositivo que estamos controlando es una cámara y tiene una propiedad llamada CCD_EXPOSURE que expresa el tiempo de exposición que queremos para una toma (en segundos), el cliente puede pedirle al *driver* que cambie dicho valor a 10.5 si queremos una toma de dicha duración. En ese momento el *driver* actualizará el valor de dicha propiedad y se lo notificará al cliente y comenzará la nueva exposición. Durante la exposición el *driver* puede mandar mensajes al cliente especificando que el tiempo de exposición cada vez es más pequeño (y además puede haber señalado que dicha propiedad está ocupada (●BUSY) (va

quedando menos tiempo para terminar la toma). Una vez finalizada la toma, la propiedad volverá al valor 0, estado OK y la propiedad de tipo *BLOB* que contiene los datos RAW de la imagen también se actualizará con la imagen recién capturada.

Para conseguir que los clientes puedan interpretar y presentar un interfaz de usuario adecuado al usuario (no es lo mismo manejar una cámara que una montura, que una rueda porta-filtros), existen una serie de propiedades estándar que deben definirse cuando desarrollamos un *driver* de una tipología concreta de dispositivo. Por ejemplo, si queremos desarrollar un *driver* para una rueda porta-filtros, debemos implementar dos propiedades: *FILTER_SLOT* (numérica), que determina cual es el filtro seleccionado en cada momento y *FILTER_NAME* (cadena de caracteres) que determina el nombre del filtro en cada hueco de la rueda.

3. INDIGO: Evolucionando INDI

INDIGO [15] aparece en 2016 como una escisión del proyecto INDI. Este nuevo proyecto presenta algunas novedades y diferencias que pueden suponer una ventaja frente al uso o desarrollo de aplicaciones bajo INDI. Entre otras cabe citar:

- **Cambios en la licencia:** INDILib (la implementación de referencia de INDI) tiene una licencia de código abierto. Concretamente utiliza la licencia GNU LGPL v2.1 [16]. Dicha licencia ofrece muchas libertades al público que quiera modificarla, usarla (incluso en proyectos remunerados), distribuirla, etc. Sin embargo hay un aspecto de dicha licencia que no gustaba a los desarrolladores principales de INDIGO. La licencia GNU LGPL exige que si la usas, el código fuente del *software* que se distribuya que haya utilizado la biblioteca debe también liberarse. Este aspecto viral de la licencia (que obliga a mantener el software que se desarrolle libre y accesible para el resto de la comunidad) "impide" en cierta manera obtener una remuneración por el *software* desarrollado según los esquemas típicos (vender la aplicación), ya que el software libre permite a cualquiera compilar el código fuente y obtener una copia funcional del software sin ningún coste ni restricción. Por tanto, la licencia de INDIGO no exige la publicación del código fuente del software que se haya desarrollado usándola (en caso de que sea un software que se distribuya).
- **Mejoras de rendimiento:** Dado que la comunicación entre los clientes, servidores y *drivers* INDI se realiza usando usando el protocolo INDI y mediante el uso de *sockets* de red o *pipes* entre procesos cuando varios de dichos componentes se encuentran en la misma máquina (lo cual es común en configuraciones simples donde no hace falta un control remoto de dispositivos) INDI es ineficiente y, de hecho, poco útil para aplicaciones que exijan una alta velocidad (como por ejemplo *lucky-imaging* o fotografía planetaria). Por ello la implementación base de INDIGO permite que la comunicación entre los componentes pueda hacerse directamente en la memoria del ordenador lo cual incrementa la eficacia en varios ordenes de magnitud. Pese a la aparente ventaja no hay que perder de vista que dicho aumento de eficiencia NO es posible en un esquema distribuido.
- **Mejoras en la transmisión de los datos binarios:** Los datos binarios (como las imágenes capturadas por las cámaras) se pueden transmitir de manera más eficiente a los clientes usando un servidor *http* que incorpora el servidor INDIGO. Además se desacopla el envío de dichos datos del resto de información (resto de propiedades) del dispositivo.
- **Incorporación de JSON al protocolo de comunicación:** Con INDIGO se puede optar por mandar la información tanto en XML (como hace INDI) como en JSON [17], un formato más actual de codificación de la información. Esto puede facilitar o hacer más cómoda la creación de componentes de INDIGO en otros lenguajes de programación como Javascript.

- **Racionalización de los nombres:** Los nombres de las propiedades estándar de INDI y de los *drivers* se han ajustado para que sean más coherentes entre sí y facilitar su utilización. En [18] se puede ver la lista de propiedades estándar para cada tipo de dispositivo astronómico existente.
- **Retrocompatibilidad:** Pese a la cantidad de cambios introducidos, INDIGO mantiene una alta compatibilidad con INDI. Es decir, es posible usar un cliente INDI para conectar con un servidor INDIGO o a la inversa. Eso implica que, por ejemplo, si aparece un nuevo *driver* INDI para un dispositivo que aún no tiene su contrapartida programada en INDIGO, es perfectamente posible usarlo en esta nueva plataforma.

Por algunas de las razones mencionadas los autores hemos elegido esta plataforma para el desarrollo de esta serie de artículos sobre la construcción y programación de dispositivos astronómicos. En la siguiente sección finalizaremos con una breve descripción del dispositivo que, a modo de ejemplo, vamos a usar para esta tarea.

4. Descripción general del dispositivo astronómico a desarrollar

Desde la aparición de INDI, poco a poco se han ido desarrollando los *drivers* para el control de muchos dispositivos comerciales, ya sean desarrollados por las propias compañías para dar soporte a esta plataforma o por desarrolladores independientes que necesitan un *driver* de control específico para un cierto dispositivo y así poder controlarlo desde su cliente.

Debido a la libertad de desarrollo de *drivers* que ofrecen tanto INDI como INDIGO, han empezado a surgir proyectos basados en microcontroladores para poder interactuar con dispositivos usados comúnmente en astronomía (motores o servomotores, *encoders*, finales de carrera, etc). Uno ejemplo bastante conocido es el de la motorización del control de foco en un telescopio con el proyecto conocido como myFocuserPro2 [13]. Éste proyecto está basado en la plataforma Arduino para construir la electrónica y *firmware* del dispositivo. Además del propio dispositivo los autores han desarrollado *drivers* tanto para la plataforma INDI como para ASCOM.

Para algunas tareas típicas en astronomía puede hacer falta el control de pequeños dispositivos como motores, servomotores, u otros actuadores que pueden ser difíciles (o caros) de obtener como soluciones comerciales (por lo particular que puede ser nuestro equipamiento astronómico). Es por ello que el movimiento "hazlo to mismo" (Do It Yourself, DIY) está cobrando bastante importancia y mucha gente desarrolla sus propios controladores de monturas, enfocadores, ruedas porta-filtros, rotadores, control de cúpulas, etc., aunque en realidad el uso de microcontroladores puede tener otras muchas aplicaciones en el ámbito de la astronomía.

En esta serie de artículos (y a modo de ejemplo) nos centraremos en crear un panel de *flats/darks* robótico (Fig. 4), que además va a poder usarse como tapadera para resguardar las ópticas mientras el equipo no esté en funcionamiento.

Nuestro panel de *flats* tendrá dos componentes electrónicos fundamentales que controlaremos con una electrónica basada en Arduino:

1. Un motor servo digital DS RDS3115MG. Permitirá abrir/cerrar la tapa que se posicionará en la entrada del telescopio.
2. Módulo de control PWM u optoacoplador ILD213T. Ésto nos permitirá controlar la luz del propio panel de *flats* independientemente del voltaje que requiera.

Debido a la gran variedad de telescopios existentes en el mercado, tendremos que adaptar nuestro panel de *flats* con piezas creadas ex profeso. Por este motivo diseñaremos las mismas mediante un software de modelado paramétrico (FreeCAD) y luego las imprimiremos con una impresora 3D.



Figura 4. Panel automático de flats en telescopio Skywatcher Esprit 120 ED.

En el siguiente artículo describiremos como podemos comenzar a construir nuestro dispositivo, partiendo desde los componentes que necesitamos, programar el *firmware* en el microcontrolador e implementar un protocolo de comunicación. En los siguientes artículos nos centraremos en la programación del driver oportuno para INDIGO y la creación de un cliente específico.

5. Conclusiones

En este primer artículo de la serie dedicada al diseño, construcción y programación de dispositivos astronómicos hemos hecho una introducción a como los sistemas operativos de nuestros ordenadores utilizan mecanismos de abstracción para ser capaces, mediante el uso de *drivers*, de controlar distintos dispositivos periféricos que conectemos a los mismos. En el caso particular de las astronomía, donde la especificidad de los dispositivos hace que los desarrolladores de sistemas operativos no incluyan soporte directo para una amplia variedad de dispositivos como pueden ser ruedas porta-filtros, enfocadores, monturas, etc, hace falta la utilización de plataformas software de más alto nivel como son ASCOM, INDI o INDIGO que permitan un control y programación de nuestro instrumental.

Hemos presentado con más detalle dos de dichas plataformas (INDI y INDIGO). Ambas están pensadas y diseñadas desde el principio teniendo en cuenta que una de las tendencias actuales en astronomía es el control remoto y distribuido del instrumental. Particularmente, en los futuros artículos de la serie haremos uso de INDIGO como plataforma de desarrollo de los *drivers* de nuestros dispositivos.

Por último, hemos realizado una somera descripción del dispositivo que, a modo de ejemplo, desarrollaremos en los siguientes artículos de la serie: un panel de *flats/darks* para nuestro telescopio.

Referencias

- [1] von Neumann, John (1945), First Draft of a Report on the EDVAC. Disponible en <https://web.archive.org/web/20130314123032/http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf>

- [2] Wikipedia - Arquitectura de von Neumann. Disponible en: https://es.wikipedia.org/wiki/Arquitectura_de_Von_Neumann
- [3] Tanenbaum, Andrew (2023), Modern Operating Systems, Global Edition. Pearson Education Limited. ISBN: 978-1292459660.
- [4] Wikipedia - Hardware abstraction. Disponible en: https://en.wikipedia.org/wiki/Hardware_abstraction
- [5] ASCOM. Astronomy Common Object Model. Disponible en: <https://ascom-standards.org/>
- [6] ASCOM Alpaca. Disponible en: <https://ascom-standards.org/AlpacaDeveloper/Index.htm>
- [7] Wikipedia. Instrument Neutral Distributed Interface. Disponible en: https://en.wikipedia.org/wiki/Instrument_Neutral_Distributed_Interface
- [8] Downey, Elwood Charles (2007), INDI: Instrument-Neutral Distributed Interface, Protocol Version 1.7, Document Version 1.3. Disponible en: <https://clearskyinstitute.com/INDI/INDI-1.7-1.3.pdf>
- [9] INDI Lib. Open Astronomy Instrumentation. Disponible en: <https://indilib.org/>
- [10] Free Software Foundation. ¿Qué es el software libre?. Disponible en: <https://www.gnu.org/philosophy/free-sw.es.html>
- [11] INDIForJava. Disponible en: <https://indiforjava.github.io/>
- [12] INDI Python Bindings. Disponible en: <https://www.indilib.org/develop/indi-python-bindings.html>
- [13] Arduino ASCOM Focuser Pro2 DIY. Disponible en: <https://sourceforge.net/projects/arduinoascomfocuserpro2diy/>
- [14] Wikipedia. Pipeline (Unix). Accesible en: [https://en.wikipedia.org/wiki/Pipeline_\(Unix\)](https://en.wikipedia.org/wiki/Pipeline_(Unix))
- [15] INDIGO Astronomy. Accesible en: <https://www.indigo-astronomy.org/>
- [16] GNU Lesser General Public License, version 2.1. Accesible en: <https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>
- [17] Wikipedia. JSON. Accesible en: <https://es.wikipedia.org/wiki/JSON>
- [18] INDIGO properties. Accesible en: https://github.com/indigo-astronomy/indigo/blob/master/indigo_docs/PROPERTIES.md

SECTION: ASTRONOMICAL COMPUTING

Cálculo de ángulos de orientación de la Tierra y transformaciones de coordenadas

Tomás Alonso Albi¹¹Starion/European Space Agency, Spain. E-mail: talonsoalbi@gmail.com.**Keywords:** programación, programming, efemérides, ephemerides, cálculo astronómico, astronomical computing

© Este artículo está protegido bajo una licencia Creative Commons Attribution 4.0 License

Este artículo adjunta un *software* accesible en <https://github.com/JCAAC-FAAE>**Resumen**

En esta entrega de la sección discutimos el cálculo de la diferencia TT–UT1, que resulta básico para cualquier cálculo de fenómenos, ya que relaciona el tiempo en el que se expresan las efemérides planetarias con el tiempo asociado a la rotación de la Tierra y, por tanto, a la posición de un observador sobre la misma. El Tiempo Sidéreo Aparente, que es esencialmente el ángulo de rotación terrestre, se relaciona de manera directa con UT1 y con la ecuación de los equinoccios. Además, se discute el cálculo de los parámetros de precesión y nutación, y de la oblicuidad de la eclíptica, necesarios para diversas y necesarias transformaciones entre sistemas de coordenadas astronómicas, que se tratan al final del artículo.

Abstract

In this article we discuss the calculation of the TT–UT1 difference, which is fundamental for any calculation of astronomical phenomena, as it relates the time in which planetary ephemerides are expressed to the time associated with the Earth's rotation and, therefore, to the position of an observer on its surface. Apparent Sidereal Time, which is essentially the Earth's rotation angle, is directly related to UT1 and the equation of the equinoxes. Additionally, the calculation of precession and nutation parameters, as well as the obliquity of the ecliptic, is discussed, as these are necessary for various essential transformations between astronomical coordinate systems, which are addressed at the end of the article.

1. Introducción

En el número anterior [1] vimos el cálculo del día juliano, un primer paso básico para la inmensa mayoría de los cálculos astronómicos, y vimos que en general utilizaríamos el día juliano en Tiempo Universal (UT). En esta segunda entrega seguiremos con el mismo criterio de representar el día juliano en UT, y avanzaremos hacia el objetivo de obtener las efemérides con el cálculo de diferentes parámetros que están ligados a la rotación y orientación del eje de la Tierra:

- Diferencia entre el Tiempo Terrestre (TT) menos el UT. Es otro parámetro esencial para el cálculo de efemérides, dado que éstas se calculan a partir del TT. Este valor está asociado a pequeñas variaciones en el ritmo de rotación de la Tierra, y no es posible prever su evolución.
- Ángulos de precesión, nutación, y oblicuidad, que definen la orientación del eje terrestre en el espacio. Como sabemos el eje terrestre no es estable en el tiempo, pues la Tierra gira como una peonza a punto a detenerse, con un periodo de casi 26000 años. Además, la Luna y los planetas generan otro movimiento menor de nutación. Esto afecta a las coordenadas aparentes de los objetos.

- Tiempo sidéreo local, que puede definirse con la coordenada de ascensión recta o longitud celeste que en un instante dado esta cruzando el meridiano local del observador, lo cual depende del instante y la longitud geográfica del observador.

El primero de los códigos presentado en esta ocasión puede también encontrarse en el repositorio de GitHub [2] correspondiente a esta sección, cuyo uso (y posible adaptación a otros lenguajes) ayudará a evitar los frecuentes despistes y errores de codificación del pasado.

Posteriormente, en esta misma, entrega utilizaremos estos ángulos para hacer todo tipo de transformaciones entre coordenadas ecuatoriales (ascensión recta y declinación), eclípticas (longitud y latitud eclípticas), u horizontales (acimut y elevación), además de corregir las coordenadas por precesión y nutación durante el proceso posterior que veremos para el cálculo de efemérides. El correspondiente código también se puede encontrar en el repositorio.

2. Cálculo de los ángulos de orientación de la Tierra

2.1. Cálculo de la diferencia TT–UT

En el primero de los listados que se muestra al final del artículo, este cálculo tiene lugar en la subrutina contenida entre las líneas 14 y 40. Inicialmente se hace uso del código de la entrega anterior para operar con el día juliano y obtener el año, mes, y día. Luego se utilizan diferentes polinomios para efectuar el cálculo en diferentes intervalos de años: un ajuste parabólico que sigue el trabajo de Stephenson & Morrison [3] para años anteriores al -500, y posteriores al 2200, y otros ajustes para los intervalos -500 al 1600, y 1600 al 2200. Los valores entre los años 2020 y 2200 son extrapolaciones tentativas y con el tiempo habrá que refinarlas. De hecho, como se puede comprobar al final del listado el cálculo propuesto devuelve casi 71 segundos, cuando en otras referencias de Internet (como las páginas web mostradas al final del código) es posible comprobar que el valor correcto ha dejado de crecer con la tendencia que seguía en años anteriores (el ajuste utilizado se hizo hace un par de años), manteniéndose por debajo de 70 segundos. Algunos trabajos de investigación sugieren que el motivo es la desaparición acelerada de los glaciares por el cambio climático, que está compensando, al menos temporalmente, la tendencia observada por Stephenson y Morrison de que el día solar medio se alarga unos 2 milisegundos por siglo, por el efecto de la influencia de la Luna.

En otros programas esta corrección se hace con más precisión mediante interpolación en tablas como la disponible en la página web de USNO [4], o del IERS (Servicio Internacional de Rotación de la Tierra y Sistemas de Referencia) [5], pero por razones de espacio aquí proporcionamos un ajuste polinómico aproximado, válido para muchos siglos. En todo caso, el efecto de este error de pocos segundos en las coordenadas de los objetos es mínimo.

En el listado se observa una corrección en las líneas 37 y 38, que dependen de valores calculados en otras líneas. Esta corrección sólo se aplica al cálculo de Morrison para épocas remotas en el pasado o futuro. En dicho trabajo se utilizó la posición de la Luna para estimar la diferencia TT–UT, en base a registros de eclipses pasados, pero se utilizó una teoría antigua para estimar la posición de la Luna, que ahora sabemos que tiene un cierto error. Para corregir este efecto se utiliza un valor más moderno de la aceleración secular de la Luna en longitud eclíptica: -25.858 segundos de arco por siglo al cuadrado, en lugar del valor antiguo de -26 .

En el servidor de efemérides del JPL [6] el cálculo de este parámetro no sigue exactamente el método de Stephenson y Morrison, sino que el valor se deriva de la integración numérica de las últimas efemérides del JPL. El siguiente código puede utilizarse en lugar del sugerido en el caso de que el lector prefiera obtener un valor más parecido al de Horizons:

1 TTminusUT = 32.498952 * Math.pow((jd - 2377032.2) / 36525.0, 2.0) - 120;
 2 double t = (jd - J2000) / 36525.0;

```

3  if (year > -720 && year < 1320) TTminusUT -= (((((((((2.6208E-8 * t + 3.12608E-6) * t
      + 1.29912E-4) * t + 1.04337E-3) * t - 0.0939895) * t - 3.74281165) * t -
      64.8913308) * t - 598.32335) * t - 2832.8992) * t - 5500.079525) * t - 715.276688;
4  if (year > 1320 && year < 2020) TTminusUT -= (((((((((-0.0144076 * t - 0.459961) * t
      - 6.1225996) * t - 43.893463) * t - 182.345180) * t - 437.489625) * t -
      561.221579) * t - 301.145054) * t + 14.885739) * t + 91.5968394) * t - 45.3217576;
5  if (year >= 2020) TTminusUT = 69.185;

```

En todo caso, el efecto en las efemérides es muy limitado, y como puede comprobarse en la Fig. 1, el algoritmo propuesto arriba aún muestra una diferencia de algunos segundos en torno al año 2000 y en siglos anteriores, diferencia que se incrementa notablemente en épocas remotas.

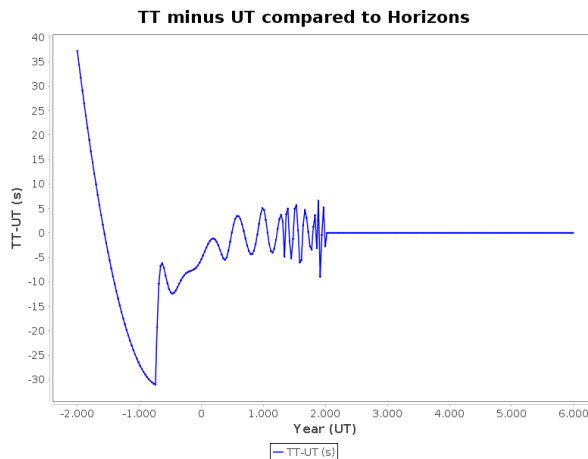


Figura 1. Diferencia entre los valores de TT–UT obtenidos por el código alternativo superior y el servidor Horizons.

2.2. Cálculo de los ángulos de nutación

La nutación es un pequeño bamboleo del eje terrestre con un periodo principal de 18.6 años, provocado por la influencia gravitatoria del Sol, la Luna y los planetas. El código de la subrutina contenida entre las líneas 60 y 75 es muy sencillo en comparación con códigos más evolucionados que pueden alcanzar precisiones mejores que el milisegundo de arco, como los algoritmos aprobados por las últimas resoluciones de la Unión Astronómica Internacional (UAI), pero para el propósito que tenemos aquí en cuanto a requerimientos de precisión, y por motivos de espacio, el algoritmo propuesto es suficiente, pues el error que produce es a lo sumo de pocas décimas de segundo de arco. El código sigue la obra de Jean Meeus [7].

En el cálculo se utiliza un método previo que permite obtener el lapso de tiempo en siglos julianos. También se hace uso de constantes que aparecen en la clase Constant, que puede consultarse en el repositorio. El nombre de las constantes, si bien se indica en inglés, debería clarificar la operación. Aparecen constantes como SECONDS_PER_DAY, que obviamente es 86400, que quizá el lector no esté habituado a utilizar, sustituyéndolas directamente por su valor. Pero es buena práctica utilizarlas para dejar más claro en el código que el objetivo es transformar las unidades de un valor de días a segundos, o al revés si se divide por ese valor, evitando así posibles errores en la operación si se introduce un número erróneo, los cuales podrían resultar muy difíciles de corregir.

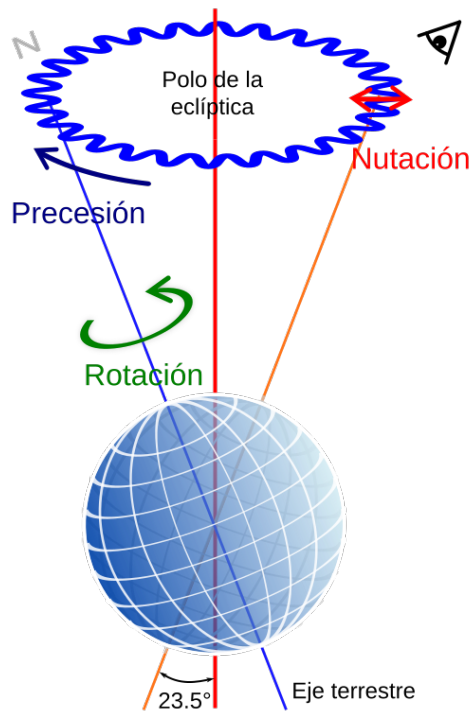


Figura 2. Movimientos de precesión y nutación del eje terrestre respecto al polo eclíptico (Wikipedia)..

2.3. Cálculo de la oblicuidad media de la eclíptica

La oblicuidad de la eclíptica es la inclinación media (sin tener en cuenta la nutación) del eje terrestre respecto a la perpendicular al plano de la eclíptica. Su cálculo se hace en la práctica con la evaluación de un polinomio que ajusta la evolución de este valor a lo largo de los siglos. Existen múltiples modos de hacer esto, y por claridad en el código se muestran tres expansiones. La más sencilla es la adoptada en las resoluciones de la UAI de 1976 [8], utilizada aún por el servidor Horizons. En la práctica nosotros también usaremos esta expansión para tratar de acercarnos lo más posible a este servidor de efemérides. La expansión de Laskar es compatible con la anterior, da esencialmente el mismo resultado, pero puede utilizarse más lejos del año 2000. La expansión de Capitaine et al. [9] es la utilizada en las resoluciones de la UAI del año 2000, y en principio no la utilizaremos, dado que implica también el uso de otros algoritmos por consistencia. Las referencias se muestran en el código.

Existen expansiones más modernas como la correspondiente al trabajo de Vondrak et al. [10], pero son algo más laboriosas de implementar y no añaden ningún valor especial al propósito de nuestros cálculos. Aunque su uso sería interesante en programas que pretendan proporcionar una representación realista del cielo nocturno muchos miles de años en el pasado.

El método presente a continuación, *trueObliquity*, calcula la oblicuidad verdadera de la eclíptica, como resultado de sumarle a la oblicuidad media la nutación en oblicuidad. Será útil en el futuro para la transformación de coordenadas eclípticas.

2.4. Cálculo de los ángulos de precesión

Los ángulos de precesión permiten corregir las coordenadas celestes por el lento cambio a lo largo de los siglos de la dirección hacia donde apunta el eje terrestre en el cielo, que hace un movimiento 90

circular completo cada 26000 años, manteniendo una inclinación igual a la oblicuidad (ver Fig. 2). Como sabemos esto hace que miles de años atrás (o en un futuro lejano), las estrellas que marcaban la dirección del polo norte celeste fueran otras, como Vega o Thuban. El algoritmo propuesto para calcular este efecto es el clásico de Lieske [11], que de nuevo es el utilizado en el servidor Horizons. El código está simplificado asumiendo que la época de partida es siempre el año 2000, lo que nos permitirá en el futuro transformar coordenadas por precesión tanto desde el año 2000 hasta otra fecha, como hacia el año 2000. Al igual que sucede con la oblicuidad y el resto de algoritmos, la UAI ha adoptado oficialmente algoritmos más evolucionados para corregir por precesión, pero en la mayoría de los servidores de efemérides se siguen utilizando algoritmos antiguos.

2.5. Cálculo del tiempo sidéreo local

La subrutina propuesta para obtener el tiempo sidéreo local es bastante completa y compatible con Horizons. En primer lugar hay que utilizar una formulación que proporciona el valor medio para Greenwich, utilizando el día juliano de la medianoche anterior en UT. El tiempo restado se añade posteriormente en otra variable, lo que aumenta la precisión del resultado respecto de usar sólo el día juliano con todos sus decimales en la formulación. El valor final se obtiene añadiendo la longitud del observador, la proyección sobre el eje de ascensión recta de la nutación en longitud eclíptica, y un par de términos de poca amplitud que dependen del nodo ascendente de la órbita de la Luna. Hay una diferencia mínima con Horizons, como puede comprobarse en su documentación [12], pero esencialmente es el mismo cálculo.

El valor de entrada es el día juliano en UT, normalmente UTC (Tiempo Universal Coordinado), pero si es posible se puede introducir el valor en UT1, corregido por el movimiento del polo terrestre, que es un parámetro imposible de prever, y que es monitorizado por el IERS y publicado en sus boletines [5]. La corrección para un instante dado es habitualmente de una fracción de segundo de tiempo, y por los requerimientos de precisión que tenemos ignoraremos este efecto. Aunque es importante tener presente que la corrección podría llegar a ser mayor, dado que la práctica de añadir segundos intercalares al UTC desaparecerá en unos pocos años.

2.6. Código

El código completo se presenta a continuación. Las últimas líneas muestran los resultados del ejemplo de partida, para el día juliano 2457407.5 y la longitud geográfica correspondiente a Madrid. El resultado es $TT-UT1 = 66.6s$, frente a los $68.2s$ que muestra Horizons. La discrepancia en el valor del tiempo sidéreo es de apenas $0.2s$. En los boletines del IERS es posible comprobar que el DUT1 (UT1-UTC) tiene un valor de $0.0447225s$ para el instante del ejemplo, valor que puede sumarse al día juliano para reducir la discrepancia a $0.15s$. En todo caso, sin esta corrección la calculadora del USNO [13] proporciona un valor coincidente con el de este programa, con una diferencia del orden del milisegundo de tiempo. El error máximo de este programa debería ser de pocas centésimas de segundo, debido a la aproximación utilizada para la nutación.

```
1 package journal;
2
3 /**
4  * Computes multiple parameters related to the orientation of the Earth: TT minus UT1,
5  * nutation angles, mean obliquity, precession angles, and local apparent sidereal
6  * time
7  */
8 public class EarthAngles {
9
10     /**
11      * Computes the difference between Terrestrial Time and Universal Time UT1
```



```

11     * @param jd The Julian day
12     * @return TT-UT1 in seconds
13     */
14     public static double TTminusUT1(double jd) {
15         JulianDay julDay = new JulianDay(jd);
16         int year = julDay.year;
17         int month = julDay.month;
18         double day = julDay.day + julDay.getDayFraction();
19
20         double TTminusUT1 = 0;
21         double ndot = -25.858, c0 = 0.91072 * (ndot + 26.0) ;
22         if (year < -500 || year >= 2200) {
23             double u = (jd - 2385800.5) / 36525.0; // centuries since J1820
24             TTminusUT1 = -20 + 32.0 * u * u;
25         } else {
26             double x = year + (month - 1 + (day - 1) / 30.0) / 12.0;
27             double x2 = x * x, x3 = x2 * x, x4 = x3 * x, x8 = x4 * x4;
28             if (year < 1600) {
29                 TTminusUT1 = 10535.328003 - 9.9952386275 * x + 0.00306730763 * x2
30                     - 7.7634069836E-6 * x3 + 3.1331045394E-9 * x4 +
31                     8.2255308544E-12 * x2 * x3 - 7.4861647156E-15 * x4 * x2 +
32                     1.936246155E-18 * x4 * x3 - 8.4892249378E-23 * x8;
33             } else {
34                 TTminusUT1 = -1027175.34776 + 2523.2566254 * x - 1.8856868491 *
35                     x2 + 5.8692462279E-5 * x3 + 3.3379295816E-7 * x4 +
36                     1.7758961671E-10 * x2 * x3 - 2.7889902806E-13 * x2 * x4 +
37                     1.0224295822E-16 * x3 * x4 - 1.2528102371E-20 * x8;
38             }
39             c0 = 0.91072 * (ndot + 25.858) ;
40         }
41         double c = -c0 * Math.pow((jd - 2435109.0) / 36525.0, 2);
42         if (year < 1955 || year > 2005) TTminusUT1 += c;
43         return TTminusUT1;
44     }
45
46     /**
47     * Transforms the input Julian day into centuries from J2000 in Terrestrial Time,
48     * to compute ephemerides
49     * @param jd Julian day
50     * @param UT True if input Julian day is in UT, otherwise TT is assumed
51     * @return Centuries from J2000 in TT
52     */
53     public static double toCenturiesRespectJ2000(double jd, boolean UT) {
54         if (UT) jd = jd + TTminusUT1(jd) / Constant.SECONDS_PER_DAY;
55         return (jd - Constant.J2000) / Constant.JULIAN_DAYS_PER_CENTURY;
56     }
57
58     /**
59     * Computes nutation in longitude and obliquity
60     * @param jd Julian day in UT
61     * @return Nutation angles in radians
62     */
63     public static double[] nutation(double jd) {
64         double t = toCenturiesRespectJ2000(jd, true);
65
66         // Compute approximate nutation
67         // Mean longitude of the ascending node of the Moon
68         double om = (125.04452 - 1934.136261 * t + 0.0020708 * t * t + t * t * t /
69             450000.0) * Constant.DEG_TO_RAD;
70         // 2 * Mean longitude of Sun
71         double l2 = 2 * (280.4665 + 36000.7698 * t) * Constant.DEG_TO_RAD;
72         // 2 * Mean longitude of Moon
73         double l2p = 2 * (218.3165 + 481267.8813 * t) * Constant.DEG_TO_RAD;

```



```

70     double nutLon = -17.20 * Math.sin(om) - 1.32 * Math.sin(l2) - 0.23 *
71         Math.sin(l2p) + 0.21 * Math.sin(2 * om);
72     double nutObl = 9.20 * Math.cos(om) + 0.57 * Math.cos(l2) + 0.1 *
73         Math.cos(l2p) - 0.09 * Math.cos(2 * om);
74     return new double[] {nutLon * Constant.ARCSEC_TO_RAD, nutObl *
75         Constant.ARCSEC_TO_RAD};
76 }
77 /**
78  * Returns the mean obliquity in radians
79  * @param jd Julian day in UT
80  * @return Mean obliquity
81  */
82 public static double meanObliquity(double jd) {
83     double t = toCenturiesRespectJ2000(jd, true);
84     // IAU 1976 formulation, still used by Horizons
85     double eps0 = 84381.448;
86     double[] pol = {-468150., -590., 181320.};
87     // J. Laskar's expansion comes from "Secular terms of classical planetary
88         theories using the results of general theory," Astronomy and Astrophysics
89         157, 59070 (1986)
90     //double eps0 = 84381.448;
91     //double[] pol = {-468093., -155., 199925., -5138., -24967., -3905., 712.,
92         2787., 579., 245. };
93     // Capitaine et al. Astronomy and Astrophysics 412, 567-586, (2003), Hilton et
94         al. 2006,
95     //double eps0 = 84381.406;
96     //double[] pol = {-468367.69, -183.1, 200340., -5760., -43400.};
97     double meanObliquity = 0;
98     for (int i=0; i<pol.length; i++) {
99         meanObliquity += pol[i] * 0.01 * Math.pow(t * 0.01, i + 1);
100     }
101     return (meanObliquity + eps0) * Constant.ARCSEC_TO_RAD;
102 }
103 /**
104  * True obliquity
105  * @param jd Julian day
106  * @return True obliquity in radians
107  */
108 public static double trueObliquity(double jd) {
109     return meanObliquity(jd) + nutation(jd)[1];
110 }
111 /**
112  * Computes the angles to correct for precession between J2000 and another date,
113  * using the method by Lieske (IAU 1976)
114  * @param jd Julian day in UT
115  * @return The output precession angles
116  */
117 public static double[] precessionAnglesFromJ2000(double jd) {
118     double t = toCenturiesRespectJ2000(jd, true);
119     double zeta = 2306.2181 * t + 0.30188 * t * t + 0.017998 * t * t * t;
120     double z = 2306.2181 * t + 1.09468 * t * t + 0.018203 * t * t * t;
121     double theta = 2004.3109 * t - 0.42665 * t * t - 0.041833 * t * t * t;
122     zeta = Util.normalizeRadians(zeta * Constant.ARCSEC_TO_RAD);
123     z = Util.normalizeRadians(z * Constant.ARCSEC_TO_RAD);
124     theta = Util.normalizeRadians(theta * Constant.ARCSEC_TO_RAD);
125     return new double[] {zeta, z, theta};
126 }

```

```

128 }
129
130 /**
131  * Returns the local apparent sidereal time
132  * @param jd Julian day of calculations, in UTC (UT1 if possible)
133  * @param obsLon Longitude of the observer in radians
134  * @return Apparent sidereal time in radians
135  */
136 public static double localApparentSiderealTime(double jd, double obsLon) {
137     // Obtain local apparent sidereal time
138     double jd0 = Math.floor(jd - 0.5) + 0.5; // previous midnight
139     double t0 = toCenturiesRespectJ2000(jd0, false); // centuries from previous
140     // midnight
141     double secs = (jd - jd0) * Constant.SECONDS_PER_DAY;
142     double gmst = ((((-6.2e-6 * t0) + 9.3104e-2) * t0) + 8640184.812866) * t0) +
143     24110.54841;
144     double msday = 1.0 + ((((-1.86e-5 * t0) + 0.186208) * t0) + 8640184.812866) /
145     (Constant.SECONDS_PER_DAY *
146     Constant.JULIAN_DAYS_PER_CENTURY));
147     gmst = (gmst + msday * secs) * 15.0 * Constant.ARCSEC_TO_RAD;
148
149     // IAU 1994 resolution C7 added two terms (dependent on the mean ascending
150     // node of the lunar orbit omega)
151     // to the equation of equinoxes, taking effect since 1997-02-27
152     double dt = toCenturiesRespectJ2000(jd, true);
153     double omega = 125.04452 - 1934.136261 * dt + 0.0020708 * dt * dt + (dt * dt *
154     dt) / 450000;
155     omega = Util.normalizeDegrees(omega);
156
157     double nutLon = nutation(jd)[0]; // First element of the array returned by
158     // nutation
159     double last = Util.normalizeRadians(
160     gmst + obsLon + nutLon * Math.cos(meanObliquity(jd))
161     + 0.00264 * Math.sin(omega) * Constant.ARCSEC_TO_RAD
162     + 0.000063 * Math.sin(2 * omega) * Constant.ARCSEC_TO_RAD
163     );
164
165     return last;
166 }
167
168 /**
169  * Test program
170  * @param args Not used
171  */
172 public static void main(String[] args) {
173     try {
174         // Input Julian day
175         JulianDay julDay = new JulianDay(2016, 1, 20);
176         double jd = julDay.getJulianDay(); // 2457407.5
177         // Input longitude for the observer in radians. 0 = Greenwich
178         double lon = -(3 + 41 / 60.0 + 18.1 / 3600.0) * Constant.DEG_TO_RAD; //
179         // -3.68836 deg
180
181         // Compute values
182         double ttMinusUt1 = TTminusUT1(jd);
183         double[] nut = nutation(jd);
184         double obl = meanObliquity(jd);
185         double last = localApparentSiderealTime(jd, lon);
186
187         // Report values
188         // To check TT-UT1:
189         // https://maia.usno.navy.mil/ser7/deltat.data
190         // https://web.archive.org/web/20220918033245/http://asa.hmnao.com/SecK/DeltaT.html

```

```

184     System.out.println("TT-UT1: "+(float) ttMinusUt1+" s");// 66.6s, 68.18 in
        Horizons
185     System.out.println("Nutation: "+Util.formatDEC(nut[0], 2)+"", "+
186         Util.formatDEC(nut[1], 2)); // -00° 00' 00.72", -00°
        00' 09.62"
187     System.out.println("Obliquity: "+Util.formatDEC(obl, 2)); // 23° 26' 13.93"
188     System.out.println("LAST: "+Util.formatRA(last, 3)); // 07h 40m 31.144s,
        07:40:31.1426 in USNO, 07 40 31.3197 in Horizons
189     System.out.println();
190
191     // Check current sidereal time with the web page (for longitude = 0):
        https://www.localsiderealtime.com/
192     jd = new JulianDay(System.currentTimeMillis()).getJulianDay(); // Julian
        day now in UTC from the computer
        double lastNow = localApparentSiderealTime(jd, 0); // For Greenwich
193
194     System.out.println("GAST now: "+Util.formatRA(lastNow, 3));
195 } catch (Exception exc) {
196     exc.printStackTrace();
197 }
198 }
199 }
200 }

```

3. Transformaciones de coordenadas

En la sección anterior hemos visto el cálculo de ciertos parámetros ligados a la orientación y rotación del eje de la Tierra, los cuales vamos a utilizar ahora para desarrollar un código que permite rotar las coordenadas entre distintos sistemas, como los de coordenadas eclípticas, horizontales, y también galácticas y supergalácticas. Como siempre el código puede encontrarse en el repositorio de GitHub [2] correspondiente a esta sección. En próximas entregas, cuando tengamos todas la herramientas previas implementadas, empezaremos a utilizarlas para el cálculo de efemérides.

El código se ha organizado para intentar que sea fácil de entender y leer. En primer lugar, se definen métodos que permiten obtener la matriz de rotación, para cierto ángulo de entrada (siempre en radianes, siguiendo el criterio de códigos anteriores), para cada uno de los ejes. Se han llamado *getRotX*, *getRotY*, y *getRotZ*. A continuación se define el método que permite multiplicar un vector, que representa las coordenadas cartesianas de un objeto (x , y , z , o bien x , y , z , v_x , v_y , v_z , incluyendo las velocidades) por una matriz de rotación. Los otros *ladrillos* que necesitamos son los dos métodos siguientes para transformar coordenadas esféricas (como ascensión recta y declinación) en cartesianas, asumiendo que la norma del vector será la unidad, y el proceso inverso. Con estas herramientas implementadas estamos en condiciones de hacer cualquier tipo de conversión de coordenadas en muy pocas líneas de código.

3.1. Coordenadas eclípticas

La conversión de coordenadas ecuatoriales a eclípticas está definida por una rotación alrededor del eje X por un ángulo igual a la oblicuidad de la eclíptica. Esto deja la coordenada de longitud inalterada. Para seguir el criterio de los resultados que ofrece el servidor Horizons se ha incluido el día juliano en los parámetros de entrada, y el ángulo por el que se rota es la oblicuidad verdadera, que incluye la nutación en oblicuidad sumada a la oblicuidad media para el instante de cálculo. Todo ello llama al código de la entrega anterior. La conversión inversa es simplemente la misma rotación con el ángulo cambiado de signo. En algunos casos las coordenadas a convertir pueden ser astrométricas J2000, no aparentes, en cuyo caso utilizaríamos la oblicuidad media para el año 2000, conocida como la constante *eps0*.

3.2. Coordenadas horizontales

La conversión de ecuatoriales a horizontales requiere, además de la fecha, de la posición geográfica del observador. En primer lugar se hace una rotación sobre el eje Z igual al tiempo sidéreo, que permite cambiar la coordenada de ascensión recta y referirla al ángulo horario. Luego se hace una rotación sobre el eje Y por un ángulo igual a 90 grados menos la latitud, para referir la coordenada de declinación a la elevación respecto a la posición del observador. Un observador situado en el polo norte vería una elevación igual a la declinación, en cuyo caso esta rotación sería por un ángulo nulo. El último paso consiste en corregir el acimut obtenido por el criterio habitual de referirlo respecto a la dirección norte, en lugar de considerar que el acimut cero se refiere al sur (dirección del meridiano en el hemisferio norte), y al hecho de que su valor aumenta cuando la ascensión recta disminuye. Esto implica la transformación a esféricas, cambiar el signo y sumar 180° y volver a las coordenadas de tipo cartesianas para devolver el resultado. Es fácil comprobar que la conversión inversa del método siguiente sigue exactamente los mismos pasos en orden contrario, invirtiendo el orden de las rotaciones y los signos de los ángulos.

3.3. Coordenadas galácticas

La conversión de coordenadas ecuatoriales a galácticas requiere conocer la dirección del centro de la galaxia, que como sabemos coincide con la constelación de *Sagitario*. En la práctica se utiliza el polo norte galáctico para las rotaciones, que se localiza en la constelación de *La Cabellera de Berenice*. En el código se muestra en unas constantes definidas al inicio como privadas y de tipo final, que significa que su valor no puede ser obtenido desde otras clases Java ni alterado, funcionando a todos los efectos como constantes disponibles sólo en esta pieza de código. La definición de la dirección precisa del polo galáctico depende del criterio utilizado, y como puede verse nosotros adoptaremos la definición del trabajo de Jia-Cheng Liu et al. (2010). El lector interesado en conocer más detalles puede acudir al artículo de los autores, disponible libremente en el portal *arXiv*, siguiendo el enlace presente en los comentarios del código.

En cuanto a la rotación, se parece mucho a la anterior, reemplazando el tiempo sidéreo por la ascensión recta del polo norte galáctico, y la latitud por su declinación. Luego hay que hacer una transformación en la coordenada de longitud para obtener la longitud galáctica final referida al punto que se toma como origen de la longitud galáctica. Para ello se le suma la longitud del nodo más 180° , si bien en el código esto se hace mediante una rotación en el eje Z. La transformación inversa es totalmente simétrica. Es fácil comprobar que la dirección galáctica (0,0) correspondiente al centro de la galaxia corresponde a las coordenadas ecuatoriales J2000 (17h 45m 37.199s, $-28^\circ 56' 10.221''$), las cuales no coinciden con la posición de la fuente Sagittarius A*, que marca la posición física del centro de la Vía Láctea (17h 45m 40.04s, $-29^\circ 00' 28.1''$). Como se explica en el artículo esto es un problema inherente a la definición clásica de las coordenadas galácticas. En el programa hemos adoptado las constantes que menciona Jia-Cheng Liu et al. para el sistema FK5, para poder comparar los resultados con otros servidores como el NED [14]. Es posible cambiar estos valores para utilizar coordenadas ICRS, o bien cambiar los valores por los que sugiere el autor, para que además de utilizar el sistema de referencia ICRF, la posición galáctica (0, 0) coincida con el centro físico. El servidor Horizons utiliza otra definición del sistema de coordenadas galáctico, lo que incrementa la confusión.

Es importante mencionar que las coordenadas ecuatoriales de entrada y salida deben ser J2000.0, referidas al equinoccio 2000. Por tanto, puede ser necesario corregirlas por precesión antes de usar estos métodos. Lo mismo sucede con las coordenadas supergalácticas. El procedimiento para corregir por precesión se muestra más adelante.

3.4. Coordenadas supergalácticas

El sistema de coordenadas supergaláctico se refiere a un plano identificado a mitad del siglo pasado en el que se concentran muchos cúmulos de galaxias según se observan desde la Vía Láctea. Su origen se eligió para cortar al plano galáctico en la constelación de *Casiopea*, mientras su polo norte se dirige hacia *Hércules*. La definición que hemos adoptado sigue el trabajo clásico de Vaucoleurs et al. (1991), con la posición del polo referida a las coordenadas galácticas, aunque en el código se referencia el trabajo posterior de Lahav et al. (2000) [15], cuya introducción puede ser interesante de leer. Como puede verse, el plano galáctico y supergaláctico son casi perpendiculares, lo que significa que todos estos cúmulos y supercúmulos de galaxias son fáciles de observar desde la Tierra, especialmente en invierno. Por ejemplo, el centro del cúmulo de Virgo se localiza en la posición supergaláctica (104° , -2°), muy cerca del plano.

La transformación de coordenadas ecuatoriales a supergalácticas exige el paso previo de transformar las coordenadas a galácticas, debido a que el polo está referido a la posición galáctica. Luego se hace una transformación similar a las anteriores, restando al final 90° a la longitud para corregir el origen. La transformación inversa es de nuevo totalmente simétrica.

En general, todas las rotaciones son conceptualmente equivalentes y pueden definirse con tres ángulos a lo sumo, por lo que resulta muy útil y limpio disponer de un código que las implemente a partir de sus elementos más básicos.

3.5. Corrección de coordenadas por precesión

Como nota al margen, no incluido en el código listado más abajo, es posible corregir las coordenadas por precesión utilizando los métodos de rotación de esta entrega junto con el método que vimos para calcular los ángulos de precesión en la entrega anterior. Siguiendo la expresión (5) del artículo de Lieske [11]. Sea pa los tres ángulos de precesión del método *EarthAngles.precessionAnglesFromJ2000*, calculados para la fecha final deseada (recordar que el índice cero se refiere al primer elemento de la lista en el lenguaje Java), y p el vector de entrada representando las coordenadas ecuatoriales cartesianas de un objeto para la época J2000, con sólo una línea de código podemos corregir por precesión.

```
1 return rotate ( rotate ( rotate ( p, getRotZ(-pa[1])), getRotY(pa[2])), getRotZ(-pa[0]));
```

Utilizaremos esto en un futuro durante el cálculo de efemérides, así como una formulación equivalente para la nutación. La transformación simétrica con los mismos ángulos (cambiados de signo y con las rotaciones en orden inverso) nos devolvería las coordenadas J2000.

3.6. Código

El código completo se presenta a continuación, con dos ejemplos al final del código que incluyen los resultados del servidor Horizons, como comparación. En el caso de las coordenadas galácticas es mejor recurrir a la calculadora presente en el servidor NED, el cual utiliza utiliza la misma definición del sistema de coordenadas galáctico y supergaláctico que hacemos en este código. En todos los casos los resultados son coincidentes con los de Horizons o el servidor NED, hasta la tercera cifra decimal mostrada. En concreto, los resultados del primer ejemplo son:

```
1 Acimut: 118.369
2 Elevation: -22.625
3 RA: -154.261
4 DEC: -35.685
5 Ecliptic longitude: -142.786
```

6 Ecliptic latitude : -23.262
7 RA: -154.261
8 DEC: -35.685
9 Galactic longitude : -45.445
10 Galactic latitude : 26.021
11 RA: -154.261
12 DEC: -35.685
13 Supergalactic longitude : 153.839
14 Supergalactic latitude : 0.253
15 RA: -154.261
16 DEC: -35.685

```
1 package journal;
2
3 /**
4  * A class to perform coordinates transformations between equatorial, ecliptic,
5  * galactic, and supergalactic systems.
6  * The orientation of the galactic plane follows Jia-Cheng Liu et al. 2010 (see
7  * http://arxiv.org/abs/1010.3773).
8  * The orientation of the supergalactic pole follows Lahav et al 2000 (see
9  * https://arxiv.org/abs/astro-ph/9809343).
10 */
11 public class CoordinateSystem {
12
13     // J2000 position of the galactic pole in FK5 system. See Jia-Cheng Liu et al.
14     // 2010,
15     // http://arxiv.org/abs/1010.3773. Distance to galactic center was set to 25830
16     // +/- 500
17     // light years (VERA collaboration, 2020)
18     // FK5
19     private static final double GALACTIC_POLE_RA_J2000 = 192.85948120833334 *
20     Constant.DEG_TO_RAD;
21     private static final double GALACTIC_POLE_DEC_J2000 = 27.128251194444445 *
22     Constant.DEG_TO_RAD;
23     private static final double GALACTIC_NODE_J2000 = 122.93191857 *
24     Constant.DEG_TO_RAD;
25     // Redefinition by Jia-Cheng Liu et al.
26     //private static final double GALACTIC_POLE_RA_J2000 = 192.90297999208332 *
27     // Constant.DEG_TO_RAD;
28     //private static final double GALACTIC_POLE_DEC_J2000 = 27.103109214444444 *
29     // Constant.DEG_TO_RAD;
30     //private static final double GALACTIC_NODE_J2000 = 123.0075021536 *
31     // Constant.DEG_TO_RAD;
32     // Galactic position of the supergalactic pole. See Lahav et al 2000, MNRAS 312
33     // 166-176,
34     // https://arxiv.org/abs/astro-ph/9809343. Definition by G. de Vaucouleurs 1991.
35     private static final double SUPER_GALACTIC_POLE_RA = 47.37 * Constant.DEG_TO_RAD;
36     private static final double SUPER_GALACTIC_POLE_DEC = 6.32 * Constant.DEG_TO_RAD;
37
38     /**
39     * Returns a 3x3 pure rotation matrix along axis X.
40     * @param angle The angle to rotate.
41     * @return The matrix.
42     */
43     public static double[][] getRotX(double angle) {
44         return new double[][] {
45             new double[] {1.0, 0.0, 0.0},
46             new double[] {0.0, Math.cos(angle), Math.sin(angle)},
47             new double[] {0.0, -Math.sin(angle), Math.cos(angle)};
48         }
49     }
50 }
51
52 /**
```

```

39  * Returns a 3x3 pure rotation matrix along axis Y.
40  * @param angle The angle to rotate.
41  * @return The matrix.
42  */
43  public static double[][] getRotY(double angle) {
44      return new double[][] {
45          new double[] {Math.cos(angle), 0.0, -Math.sin(angle)},
46          new double[] {0.0, 1.0, 0.0},
47          new double[] {Math.sin(angle), 0.0, Math.cos(angle)}};
48  }
49
50  /**
51  * Returns a 3x3 pure rotation matrix along axis Z.
52  * @param angle The angle to rotate.
53  * @return The matrix.
54  */
55  public static double[][] getRotZ(double angle) {
56      return new double[][] {
57          new double[] {Math.cos(angle), Math.sin(angle), 0.0},
58          new double[] {-Math.sin(angle), Math.cos(angle), 0.0},
59          new double[] {0.0, 0.0, 1.0}};
60  }
61
62  /**
63  * Multiplication of a vector with a matrix
64  * @param p The vector, with 3 (position) or 6 (position and velocity) components
65  * @param m The 3x3 rotation matrix
66  * @return The result of the rotation
67  */
68  public static double[] rotate(double[] p, double[][] m) {
69      double[] out = new double[p.length];
70      for (int i=0; i<p.length; i++) {
71          int ip = i;
72          if (i > 2) ip = i - 3;
73          out[i] = 0;
74          for (int j=0; j<3; j++) {
75              out[i] += m[ip][j] * p[j];
76          }
77      }
78      return out;
79  }
80
81  /**
82  * Transform coordinates from cartesian, x y z, to spherical, lon lat r
83  * @param p x, y, z
84  * @return lon, lat (radians), r
85  */
86  public static double[] cartesianToSpherical(double[] p) {
87      double lon, lat;
88
89      double x = p[0], y = p[1], z = p[2];
90      if (y != 0.0 || x != 0.0) {
91          double h = Math.hypot(x, y);
92          lon = Math.atan2(y, x);
93          lat = Math.atan2(z, h);
94      } else {
95          lon = 0.0;
96          lat = Constant.PI_OVER_TWO;
97          if (z < 0.0) lat = -lat;
98      }
99      double rad = Math.sqrt(x * x + y * y + z * z);
100
101      return new double[] {lon, lat, rad};
102  }
103

```

```

104  /**
105  * Transforms coordinates from spherical, lon, lat, to cartesian, x y z
106  * @param lon Longitude in radians
107  * @param lat Latitude in radians
108  * @return x y z coordinates, with unity as the norm of the vector
109  */
110  public static double[] sphericalToCartesian(double lon, double lat) {
111      double rad = 1;
112      double cl = Math.cos(lat);
113      double x = rad * Math.cos(lon) * cl;
114      double y = rad * Math.sin(lon) * cl;
115      double z = rad * Math.sin(lat);
116
117      return new double[] { x, y, z };
118  }
119
120  /**
121  * Transforms coordinates from equatorial to ecliptic
122  * @param p Rectangular coordinates
123  * @param jd Julian day
124  * @return Ecliptic rectangular coordinates
125  */
126  public static double[] equatorialToEcliptic(double[] p, double jd) {
127      return rotate(p, getRotX(EarthAngles.trueObliquity(jd)));
128  }
129
130  /**
131  * Transforms coordinates from ecliptic to equatorial
132  * @param p Rectangular coordinates
133  * @param jd Julian day
134  * @return Equatorial rectangular coordinates
135  */
136  public static double[] eclipticToEquatorial(double[] p, double jd) {
137      return rotate(p, getRotX(-EarthAngles.trueObliquity(jd)));
138  }
139
140  /**
141  * Transforms coordinates from equatorial to horizontal
142  * @param p Rectangular coordinates
143  * @param jd Julian day
144  * @param obsLon Observer's longitude in radians
145  * @param obsLat Observer's latitude in radians
146  * @return Horizontal rectangular coordinates
147  */
148  public static double[] equatorialToHorizontal(double[] p, double jd, double
149      obsLon, double obsLat) {
150      double[] out = rotate(rotate(p,
151          getRotZ(EarthAngles.localApparentSiderealTime(jd, obsLon))),
152          getRotY((Constant.PI_OVER_TWO - obsLat)));
153      out = cartesianToSpherical(out);
154      return sphericalToCartesian(Math.PI - out[0], out[1]);
155  }
156
157  /**
158  * Transforms coordinates from horizontal to equatorial
159  * @param p Rectangular coordinates
160  * @param jd Julian day
161  * @param obsLon Observer's longitude in radians
162  * @param obsLat Observer's latitude in radians
163  * @return Equatorial rectangular coordinates
164  */
165  public static double[] horizontalToEquatorial(double[] p, double jd, double
166      obsLon, double obsLat) {
167      double[] in = cartesianToSpherical(p);
168      in = sphericalToCartesian(Math.PI - in[0], in[1]);

```



```

165     return rotate(rotate(in, getRotY(-(Constant.PI_OVER_TWO - obsLat))),
166                   getRotZ(-EarthAngles.localApparentSiderealTime(jd, obsLon)));
167 }
168 /**
169  * Transforms coordinates from J2000 equatorial to galactic
170  * @param p Rectangular coordinates
171  * @return Galactic rectangular coordinates
172  */
173 public static double[] equatorialJ2000ToGalactic(double[] p) {
174     return rotate(rotate(rotate(p, getRotZ(GALACTIC_POLE_RA_J2000)),
175                   getRotY(Constant.PI_OVER_TWO - GALACTIC_POLE_DEC_J2000)),
176                   getRotZ(Math.PI-GALACTIC_NODE_J2000));
177 }
178 /**
179  * Transforms coordinates from galactic to J2000 equatorial
180  * @param p Rectangular coordinates
181  * @return Equatorial rectangular coordinates
182  */
183 public static double[] galacticToEquatorialJ2000(double[] p) {
184     return rotate(rotate(rotate(p, getRotZ(-(Math.PI - GALACTIC_NODE_J2000))),
185                   getRotY(-(Constant.PI_OVER_TWO - GALACTIC_POLE_DEC_J2000))),
186                   getRotZ(-GALACTIC_POLE_RA_J2000));
187 }
188 /**
189  * Transforms coordinates from J2000 equatorial to supergalactic
190  * @param p Rectangular coordinates
191  * @return Supergalactic rectangular coordinates
192  */
193 public static double[] equatorialJ2000ToSupergalactic(double[] p) {
194     return rotate(rotate(rotate(equatorialJ2000ToGalactic(p),
195                   getRotZ(SUPER_GALACTIC_POLE_RA)),
196                   getRotY(Constant.PI_OVER_TWO - SUPER_GALACTIC_POLE_DEC)),
197                   getRotZ(Constant.PI_OVER_TWO));
198 }
199 /**
200  * Transforms coordinates from supergalactic to J2000 equatorial
201  * @param p Rectangular coordinates
202  * @return J2000 equatorial rectangular coordinates
203  */
204 public static double[] supergalacticToEquatorialJ2000(double[] p) {
205     return galacticToEquatorialJ2000(rotate(rotate(rotate(p,
206                   getRotZ(-Constant.PI_OVER_TWO)),
207                   getRotY(-(Constant.PI_OVER_TWO - SUPER_GALACTIC_POLE_DEC))),
208                   getRotZ(-SUPER_GALACTIC_POLE_RA)));
209 }
210 /**
211  * Test program
212  * @param s Not used
213  */
214 public static void main(String[] s) {
215     try {
216         /*
217         Example data from Horizons, for galactic better test from
218         https://ned.ipac.caltech.edu/forms/calculator.html
219         Date__(UT)__HR:MN R.A.__(a-appar)_DEC. Azi____(a-app)___Elev L_Ap_Sid_Time
220         TDB-UT ObsEcLon ObsEcLat GlxLon GlxLat L_Ap_Hour_Ang
221         2016-Jan-20 00:00 m 205.73877 -35.68542 118.369371 -22.625219 7.6753665728
222         68.184474 217.2137137 -23.2621310 314.366606 26.140141 -6.040551182
223         2000-Jan-20 00:00 m 31.19459 -13.82278 255.493146 -4.310376 7.6670225044
224         64.184432 23.8036275 -24.7875597 178.724053 -68.320060 5.587383271

```

```

219 */
220 double jd = 2457407.5;
221 double obsLon = -3.6879 * Constant.DEG_TO_RAD;
222 double obsLat = 40.408414 * Constant.DEG_TO_RAD;
223 double[] eq = sphericalToCartesian(205.73877 * Constant.DEG_TO_RAD,
    -35.68542 * Constant.DEG_TO_RAD);
224 // Uncomment for a second test
225 //jd = 2451563.5;
226 //eq = sphericalToCartesian(31.19459 * Constant.DEG_TO_RAD, -13.82278 *
    Constant.DEG_TO_RAD);
227
228 // Equatorial to horizontal
229 double[] hor = equatorialToHorizontal(eq, jd, obsLon, obsLat);
230 double[] shor = cartesianToSpherical(hor);
231 System.out.println("Acimut: " + Util.formatValue(shor[0] *
    Constant.RAD_TO_DEG, 3));
232 System.out.println("Elevation: " + Util.formatValue(shor[1] *
    Constant.RAD_TO_DEG, 3));
233 // Invert back to equatorial
234 eq = horizontalToEquatorial(hor, jd, obsLon, obsLat);
235 double[] seq = cartesianToSpherical(eq);
236 System.out.println("RA: " + Util.formatValue(seq[0] * Constant.RAD_TO_DEG,
    3));
237 System.out.println("DEC: " + Util.formatValue(seq[1] * Constant.RAD_TO_DEG,
    3));
238 // To ecliptic, using the ecliptic of date (true obliquity), as in the
    above results from Horizons
239 double[] ecl = equatorialToEcliptic(eq, jd);
240 double[] secl = cartesianToSpherical(ecl);
241 System.out.println("Ecliptic longitude: " + Util.formatValue(secl[0] *
    Constant.RAD_TO_DEG, 3));
242 System.out.println("Ecliptic latitude: " + Util.formatValue(secl[1] *
    Constant.RAD_TO_DEG, 3));
243 eq = horizontalToEquatorial(hor, jd, obsLon, obsLat);
244 seq = cartesianToSpherical(eq);
245 System.out.println("RA: " + Util.formatValue(seq[0] * Constant.RAD_TO_DEG,
    3));
246 System.out.println("DEC: " + Util.formatValue(seq[1] * Constant.RAD_TO_DEG,
    3));
247 // To galactic
248 //eq = sphericalToCartesian(192.85948120833334 * Constant.DEG_TO_RAD,
    27.128251194444445 * Constant.DEG_TO_RAD);
249 double[] gal = equatorialJ2000ToGalactic(eq);
250 double[] sgal = cartesianToSpherical(gal);
251 System.out.println("Galactic longitude: " + Util.formatValue(sgal[0] *
    Constant.RAD_TO_DEG, 3));
252 System.out.println("Galactic latitude: " + Util.formatValue(sgal[1] *
    Constant.RAD_TO_DEG, 3));
253 eq = galacticToEquatorialJ2000(gal);
254 seq = cartesianToSpherical(eq);
255 System.out.println("RA: " + Util.formatValue(seq[0] * Constant.RAD_TO_DEG,
    3));
256 System.out.println("DEC: " + Util.formatValue(seq[1] * Constant.RAD_TO_DEG,
    3));
257 // To supergalactic
258 double[] supergal = equatorialJ2000ToSupergalactic(eq);
259 double[] ssupergal = cartesianToSpherical(supergal);
260 System.out.println("Supergalactic longitude: " +
    Util.formatValue(ssupergal[0] * Constant.RAD_TO_DEG, 3));
261 System.out.println("Supergalactic latitude: " +
    Util.formatValue(ssupergal[1] * Constant.RAD_TO_DEG, 3));
262 eq = supergalacticToEquatorialJ2000(supergal);
263 seq = cartesianToSpherical(eq);
264 System.out.println("RA: " + Util.formatValue(seq[0] * Constant.RAD_TO_DEG,
    3));

```

```

265         System.out.println("DEC: " + Util.formatValue(seq[1] * Constant.RAD_TO_DEG,
266             3));
267     } catch (Exception exc) {
268         exc.printStackTrace();
269     }
270 }

```

References

- [1] *Astronomical computing: Cálculo del día juliano*, T. Alonso Albi, JCAAC **1**, 57 (2004).
- [2] *Repositorio de código en GitHub*, <https://github.com/JCAAC-FAAE>
- [3] *Historical values of the Earth's clock error deltaT and the calculation of eclipses*, Stephenson and Morrison (2004), <https://adsabs.harvard.edu/full/2004JHA....35..327M>
- [4] <https://maia.usno.navy.mil/ser7/deltat.data>
- [5] <https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html>
- [6] *Horizons*, disponible en [https://ssd.jpl.nasa.gov/horizons/app.html/](https://ssd.jpl.nasa.gov/horizons/app.html#/)
- [7] *Astronomical Algorithms*, Jean Meeus (Editorial Atlantic Books, 1998)
- [8] *Expressions for the precession quantities based upon the IAU (1976) system of astronomical constants*, J. H. Lieske et al., A&A **58**, 1 (1977).
<https://adsabs.harvard.edu/full/1977A%26A....58....1L>
- [9] *Expressions for IAU 2000 precession quantities*, N. Capitaine et al., A&A **412**, 567 (2003).
- [10] *New precession expressions, valid for long time intervals*, J. Vondrák et al., A&A **534**, A22 (2011).
<https://www.aanda.org/articles/aa/pdf/2011/10/aa17274-11.pdf>
- [11] *Precession matrix based on IAU (1976) system of astronomical constants*, J. H. Lieske, A&A **73**, 282 (1979). <https://adsabs.harvard.edu/full/1979A%26A....73..282L>
- [12] <https://ssd.jpl.nasa.gov/horizons/manual.html>
- [13] <https://aa.usno.navy.mil/data/siderealtime>
- [14] <https://ned.ipac.caltech.edu/forms/calculator.html>
- [15] <https://academic.oup.com/mnras/article/312/1/166/984983>

SECTION: SOFTWARE FOR PHOTOMETRY & ASTROMETRY

Astrometría con Tycho

Ramón Naves¹ and Montse Campàs²

¹Obs. Montcabrer - MPC 213, Cabriils–Barcelona, Spain. E-mail: ramonnavesnogues@gmail.com.

²Obs. Montcabrer - MPC 213, Cabriils–Barcelona, Spain. E-mail: mcampast@gmail.com.

Keywords: astrometría, fotometría, Tycho

© Este artículo está protegido bajo una licencia Creative Commons Attribution 4.0 License

Resumen

Se discuten las utilidades del software Tycho, un excelente programa para hacer astrometría y fotometría de asteroides, cometas, variables, etc. Presentamos algunos ejemplos de uso y diversas técnicas de análisis.

Abstract

Tools and procedures of the Tycho software are discussed. Tycho is an excellent program to perform astrometry and photometry for asteroids, comets, variable stars, etc. We present some examples where different analysis tools are employed.

1. Introducción

Tycho es un excelente programa para hacer astrometría y fotometría de asteroides, cometas, variables, etc. Desarrollado por Daniel Parrot, es un software que utiliza el apilamiento rápido de imágenes con soporte de GPU y seguimiento sintético (*synthetic tracking*) para mejorar la detección y observación de objetos celestes. Se trata de un software de pago, con actualizaciones constantes (algunas de ellas con un coste adicional). Su versión estándar tiene un precio de 25\$, mientras que la versión Pro cuesta 50\$. Existe sin embargo una versión de prueba, con la que se pueden probar sus utilidades y modos de operación.

En el artículo anterior de esta sección [1] hablamos de *Astrometrica*, un programa muy fácil de usar pero que ha quedado un tanto obsoleto. Tycho permite manejar imágenes de tamaño muy superior a las de *Astrometrica*. Con la llegada de las cámaras CCD y CMOS de gran formato y con cada vez más resolución, hacía falta un software más actualizado pues a *Astrometrica* se le ‘atragantan’ las imágenes pesadas. Con todo, *Astrometrica* es un programa excelente para empezar en el mundo de la astrometría.

Por otro lado, Tycho ha incorporado unas herramientas de búsqueda automática totalmente novedosas hasta ahora. Con esta herramienta es posible descubrir nuevos asteroides, NEOs e incluso cometas débiles, que ni siquiera aparecen en imágenes individuales. Gracias a este software algunos aficionados están haciendo descubrimientos que habrían sido realmente difíciles de conseguir con otro tipo de software.

En su última versión, a petición del Dr. Mark Kidger y del grupo de Observadores de Cometas, se ha incluido el cálculo de fotometría de cometas *multibox*, y el cálculo *Afho*, muy utilizado por este grupo de observadores. Explicaremos cómo utilizar esta parte de fotometría cometaria en un próximo artículo de esta sección.

2. Cómo configurar Tycho

Lo primero que necesitamos hacer es descargar el programa Tycho, que encontraremos en el enlace

<https://www.tycho-tracker.com/download>

Para instalar el programa, descomprimos los tres archivos de los que se compone la distribución en nuestro disco duro. La primera se refiere a la configuración (*setup*). Si vamos al menú **Settings**, comenzaremos cargando bases de datos en **Known Objects**. Si queremos hacer astrometría de cometas y asteroides, clicaremos sobre **Download MPCORB.DAT** y sobre **Download ELEMENTS.COMET** para descargar las bases de datos de asteroides y cometas (Fig. 1). Conviene hacerlo de forma periódica para mantenerla actualizada.

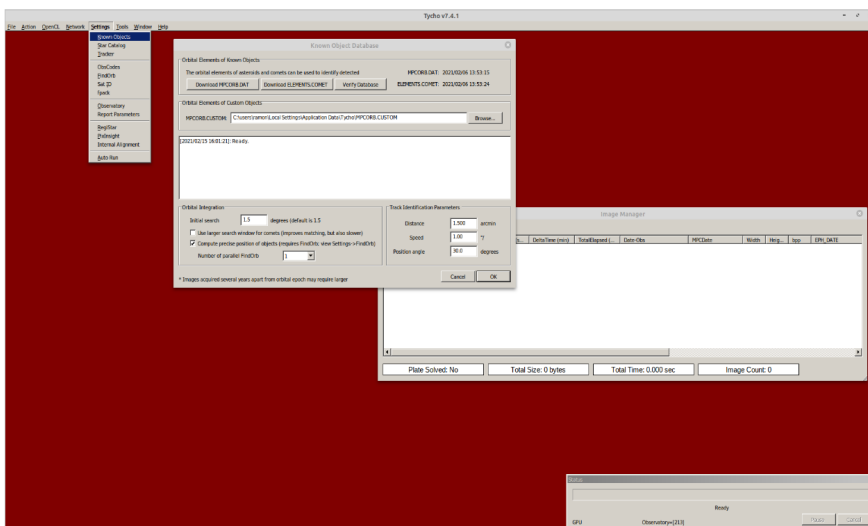


Figura 1. Menú **Settings** > **Known Objects**.

2.1. Star Catalog

Recomendamos marcar los tres cuadraditos blancos, además de **Use Online (VizieR) Access**, y en **Catalog** seleccionar **Gaia DR2** (Fig. 2).

2.2. ObsCodes

Seleccionamos el archivo **ObsCodes.html**, que encontraremos dentro de la carpeta de **Findorb** que hemos copiado en el disco duro al descargarnos el programa (Fig. 3).

2.3. FindOrb

Seleccionamos los archivos **find_o64.exe**, **find_o64-modified.exe** y **fo.exe**, que encontraremos dentro de la misma carpeta **Findorb**.

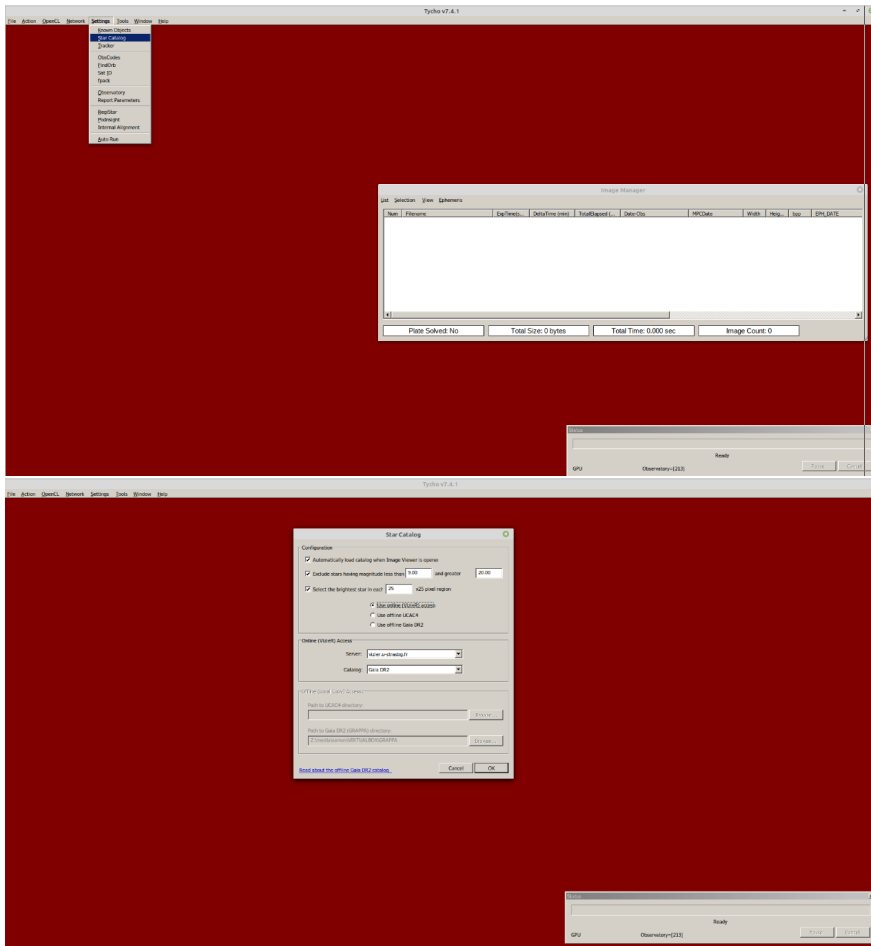


Figura 2. Menú Settings > Star Catalog.

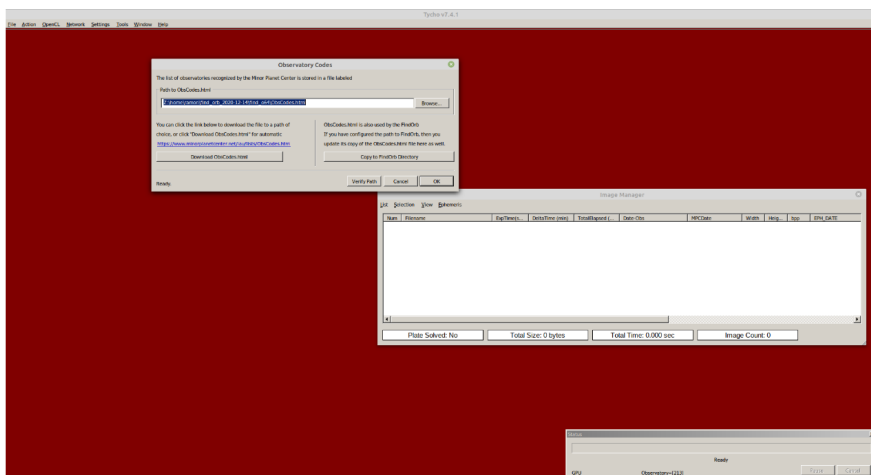


Figura 3. Menú Settings > ObsCodes.

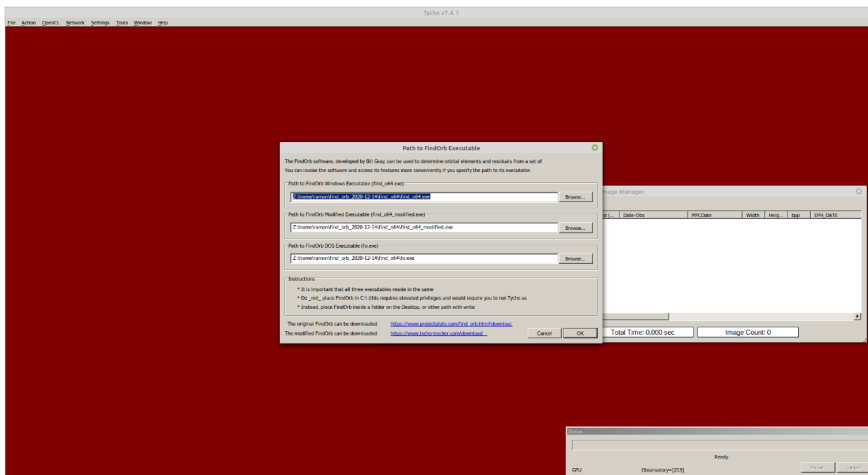


Figura 4. Menú Settings > FindOrb.

2.4. Observatory

En este submenú (Fig. 5) clicamos en Action Add Observatory, y nos aparecerá un desplegable donde pondremos nuestros datos.

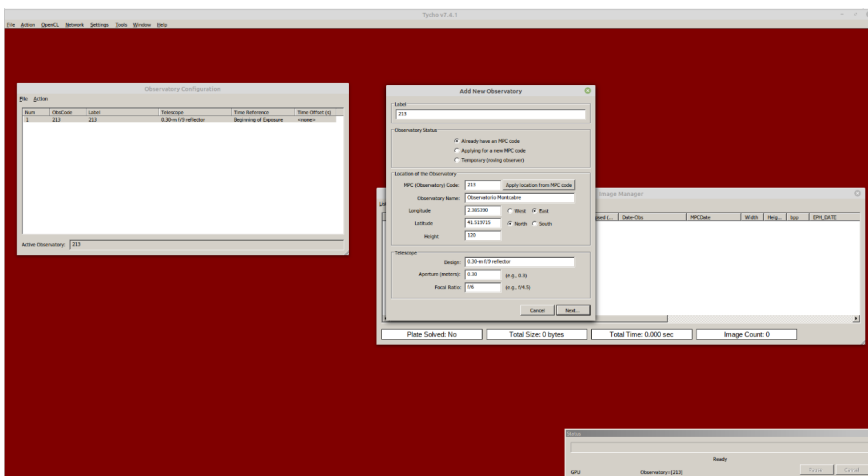


Figura 5. Menú Settings > Observatory.

2.5. Report Parameters

En este desplegable (Fig. 6) colocaremos nuestros datos personales y correo electrónico.

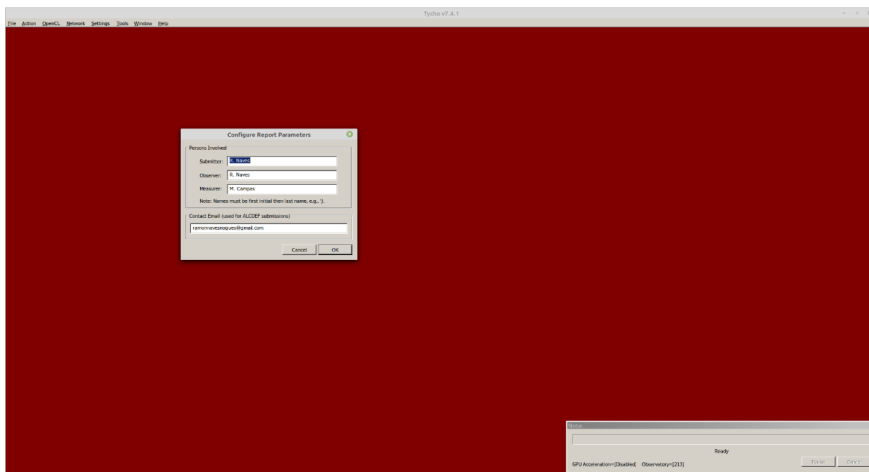


Figura 6. Menú Settings > Report Parameters.

2.6. OpenCL

En este desplegable del menú principal (Fig. 7) marcaremos el máximo número de núcleos de proceso que tenga nuestro ordenador y, si dispone de tarjeta gráfica potente con OPENCL, la seleccionamos marcando Check here to enable GPU acceleration. Es recomendable chequear el funcionamiento de nuestra tarjeta gráfica. Para ello clicamos Test Device para comprobar su compatibilidad. Ojo, *no es imprescindible* disponer de tarjeta grafica compatible con OPENCL, pero sí acelera bastante el proceso. En caso de error simplemente la desmarcaremos.

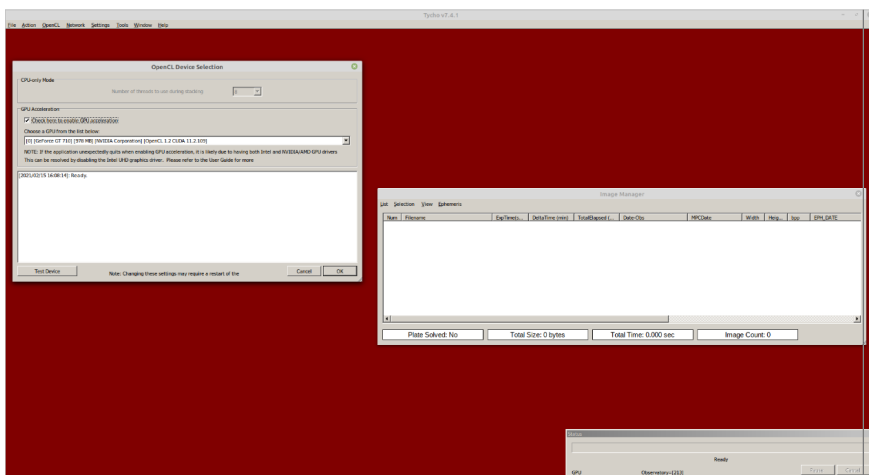


Figura 7. Menú OpenCL.

3. Preanalysis astrométrico

Clicando sobre Action se abrirá un menú, en el que seleccionaremos Express Mode (Fig. 8), y donde marcaremos los siguientes cuadraditos: Calibrate, Align, Plate solve, y Clean Up intermediate directories.

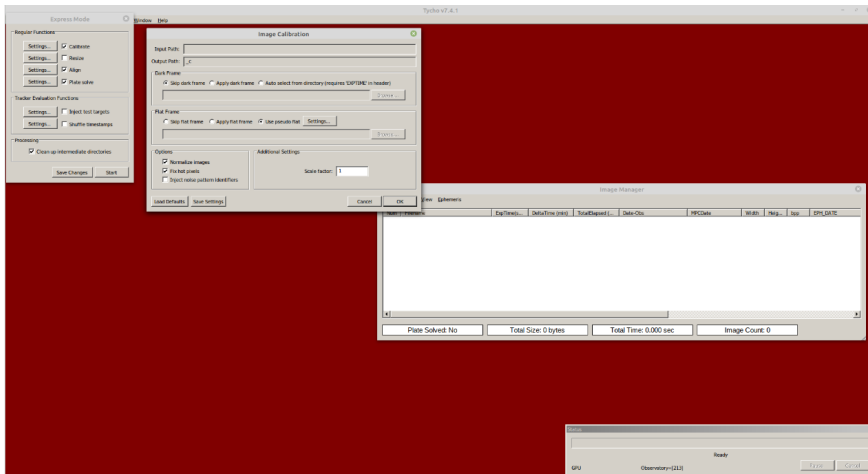


Figura 8. Menú Action > Express Mode.

3.1. Calibrate

Clicamos sobre Settings, junto a Calibrate (Fig. 8). Los autores usamos imágenes ya calibradas con sus *darks* y *flats*, pero recomendamos marcar la opción Use pseudo flat, pues aplica además unos *flats* sintéticos que permiten contrastar mucho las imágenes.

3.2. Align

Clicaremos sobre Settings, junto a Align. Los autores usamos simplemente la opción Internal Plate Solve. Clicamos ahora sobre Settings, junto a Plate Solve (Fig. 9). Recomendamos dejarlo como en el ejemplo de la imagen. No es necesario poner nuestra resolución por píxel. Configurado de esta manera resolverá prácticamente casi cualquier imagen.

4. Preanálisis de imágenes

Hacemos los siguientes pasos:

1. En el menú Image Manager clicamos en List, y seleccionamos las imágenes consecutivas, obtenidas por nosotros del cometa o asteroide que queramos analizar.
2. Clicamos en el menú principal Actions y marcamos el menú Express Mode
3. En el desplegable Express Mode, clicamos Start.

Estas acciones arrancarán un proceso que calibrará las imágenes (aplicándoles un *pseudo flat*), las alineará y por último las astrometrizará, reconociendo el campo de las mismas (incluso aunque no tengan coordenadas en las cabeceras de los ficheros *fits*). Este último proceso se demora un tiempo, dependiendo del número de imágenes, de la potencia de nuestro ordenador y/o la tarjeta gráfica utilizada.

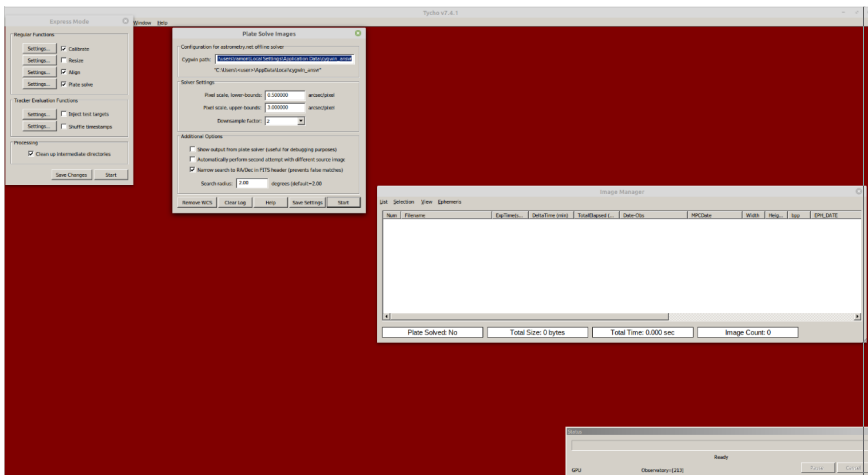


Figura 9. Menú Action > Express Mode > Settings (Plate solve).

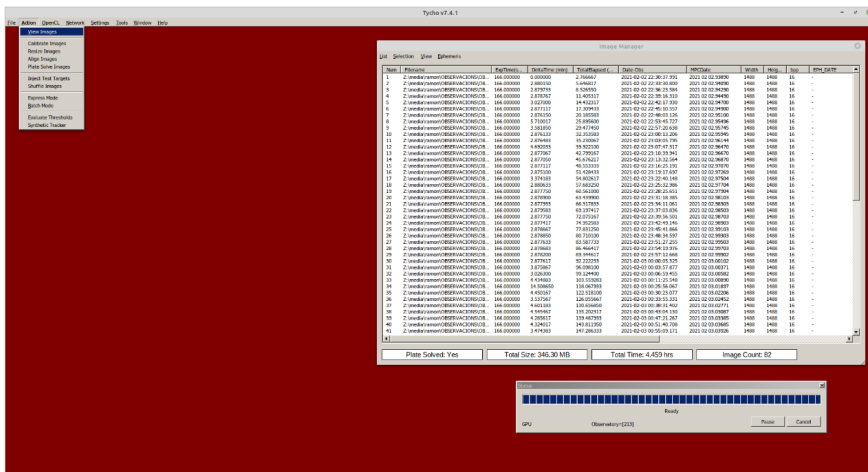


Figura 10. Menú para seleccionar y examinar imágenes.

El proceso termina cuando en la ventana Image Manager desaparece la lista de imágenes que habíamos seleccionado.

5. Obtención de resultados astrométricos

Justo después de realizar el preanálisis con Express Mode, y si todo ha ido bien, debe aparecer una carpeta nueva justo al lado de la que contenía las imágenes seleccionadas. Esta carpeta tendrá añadido el prefijo _c_a. Entonces:

1. Clicamos en el menú List de la ventana Image Manager para proceder a cargar dichas imágenes. Fijémonos que en Plate Solved: Yes ponga realmente Yes; de lo contrario, significará que el proceso no ha podido realizarse por algún motivo.
2. Clicamos en el menú Actions de la barra principal y marcamos View Images, Fig. 10.

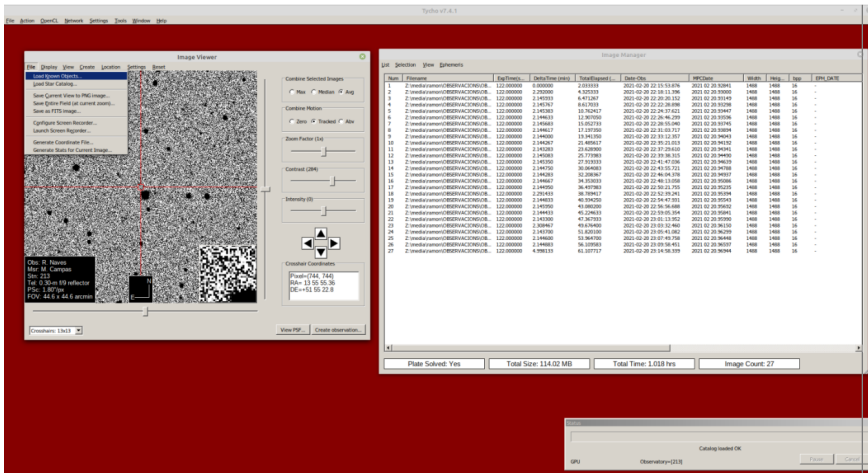


Figura 11. Menú Load Known Objects, para buscar los objetos en la imagen.

Clicamos ahora en el menú File dentro de Image Viewer, y marcamos en Load Known Objects, Fig. 11. Esto hace que el programa busque en su base de datos de cometas y asteroides qué objetos conocidos deberían estar en el campo de la imagen. A partir de aquí aparecerá una lista con esos objetos conocidos que pueden aparecer en nuestro campo de la imagen. En el caso que presentamos sólo hay un objeto, el cometa C/2021 A1. Clicamos ahora sobre la línea que nos ha aparecido en Known objects, y directamente en Image Viewer nos aparecerá una imagen apilada sobre el cometa seleccionado, Fig. 12.

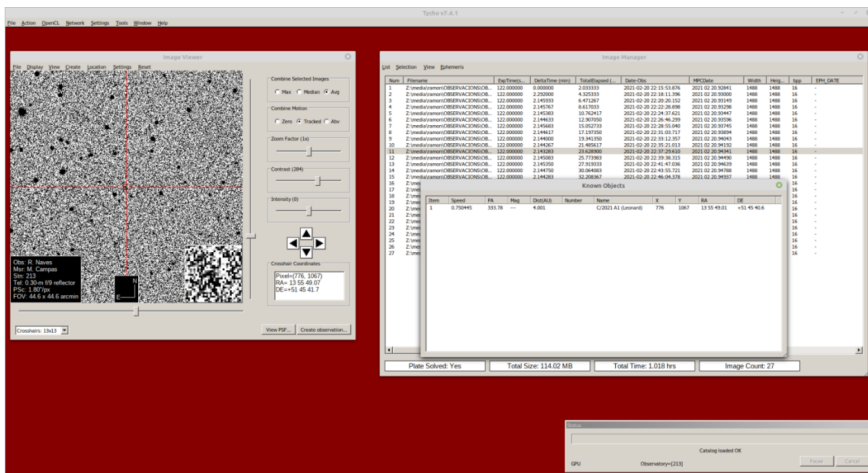


Figura 12. Imagen con la lista de objetos conocidos presentes en el campo.

Para verificar que realmente no se trata de ningún error (rayo cósmico, ruido u otro tipo de artefactos), clicaremos en Add Track Navigator, y luego sobre la línea que nos aparece en el menú Track Navigator y en Verify Track, Fig. 13. Dentro de la ventana Verify Track marcamos el cuadro Follow target, y seleccionamos el número de stacks y observaciones que deseamos, tres en este ejemplo. Veremos entonces una animación donde el cometa o asteroide se mueve, verificando que es

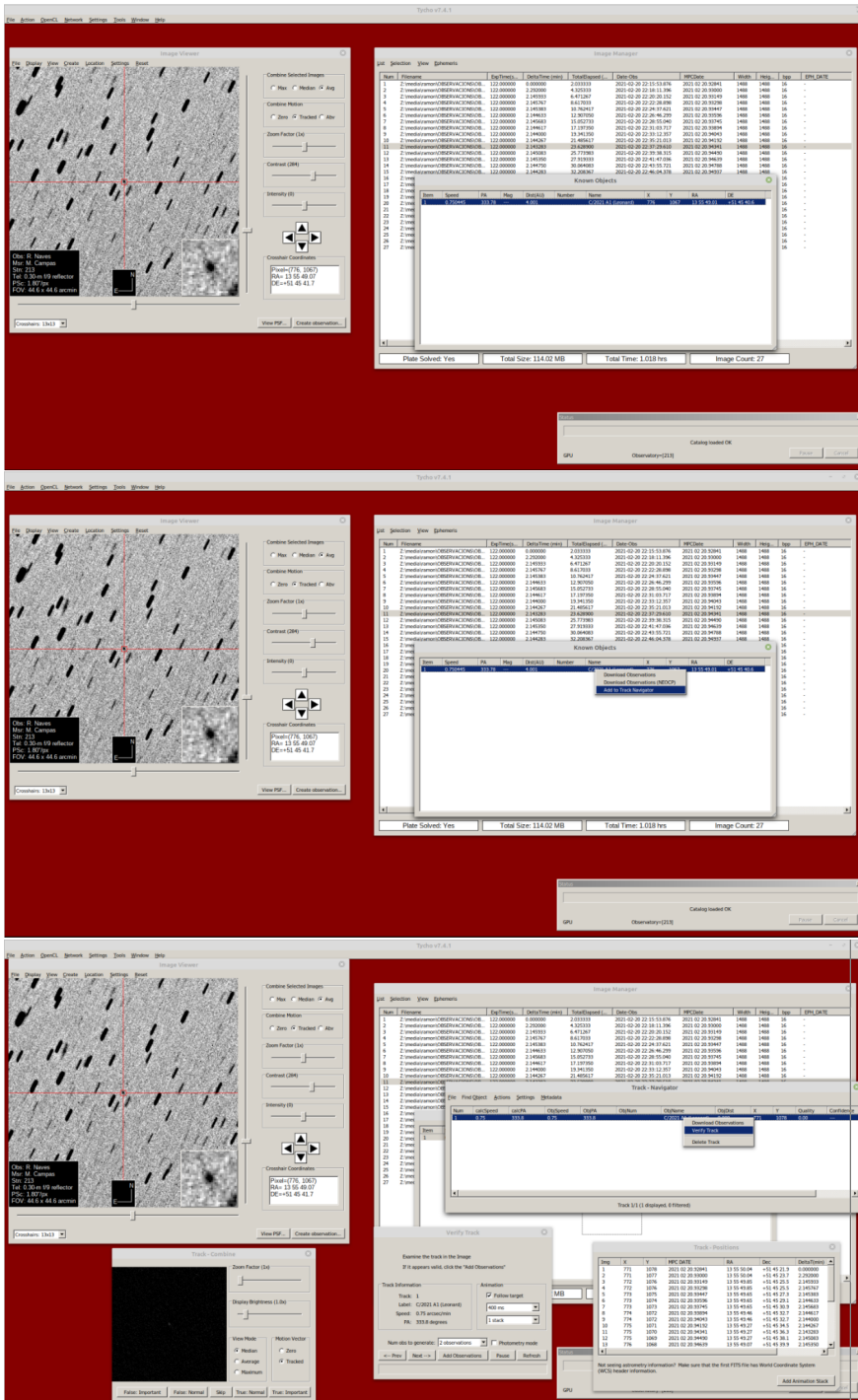


Figura 13. Ventanas de confirmación de la identidad del cometa o asteroide..

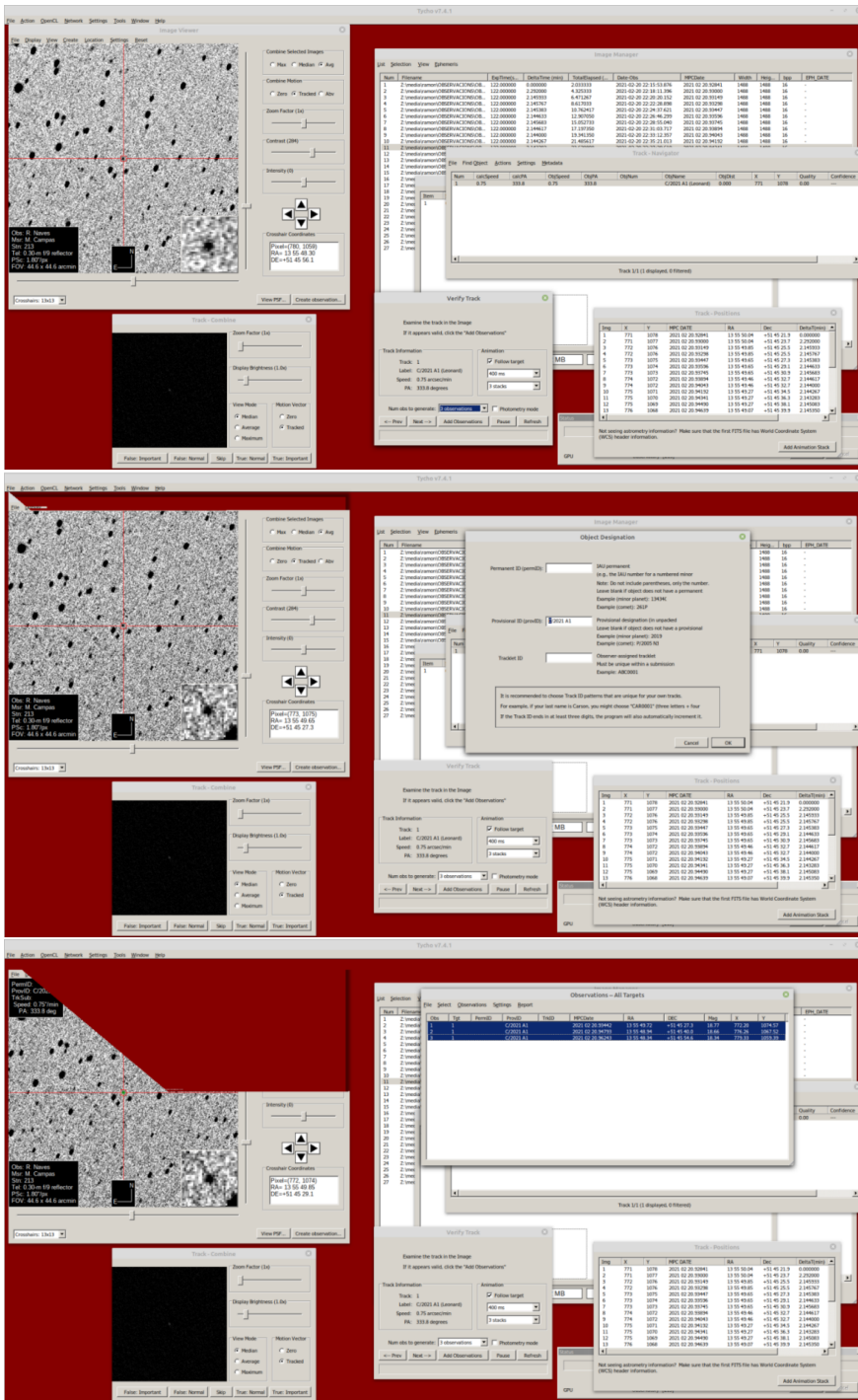


Figura 14. Ventanas de grabación de la observación y asignación de un nombre al objeto.

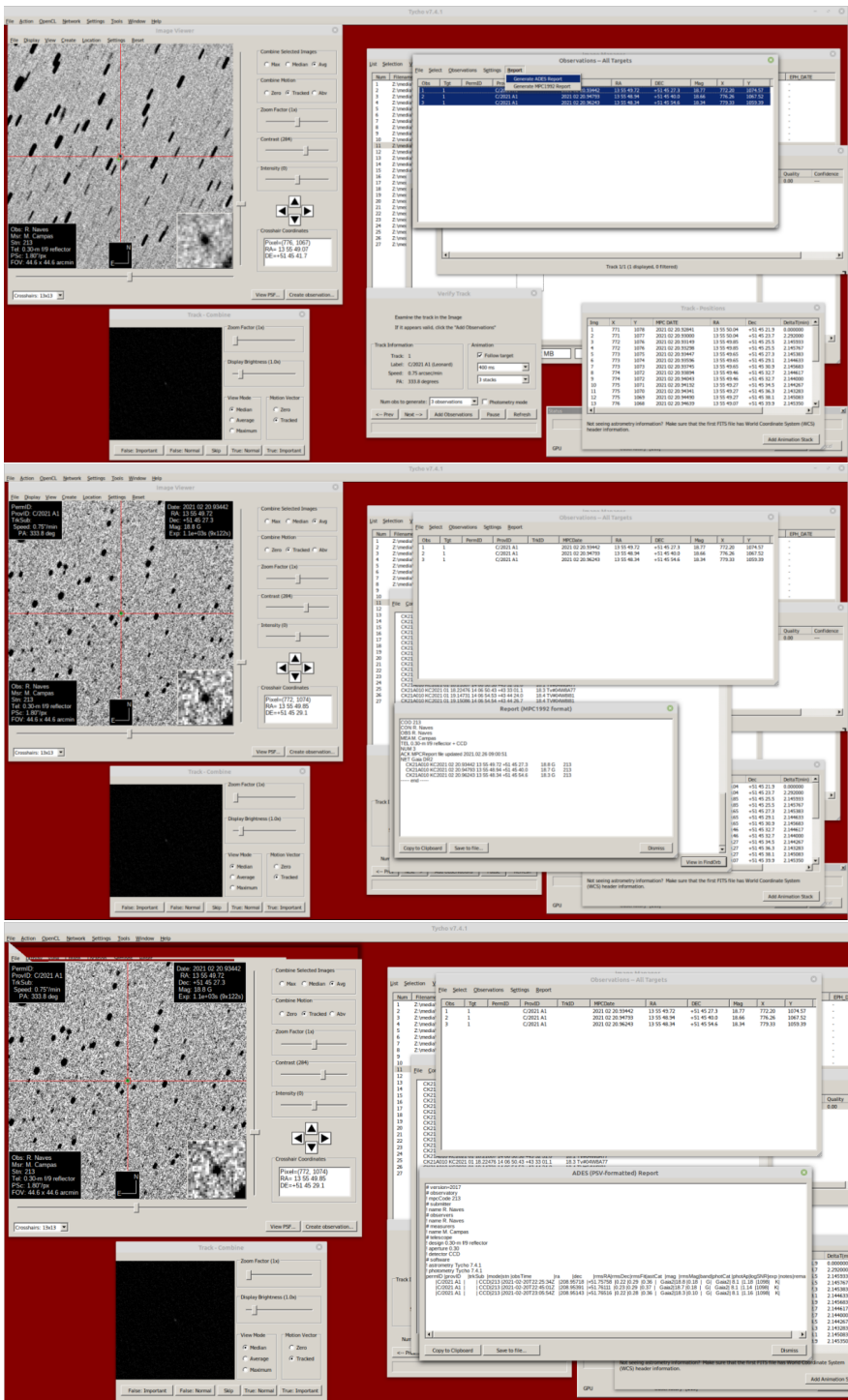


Figura 15. Ventanas sobre la generación de un informe de astrometría y su envío por correo electrónico al MPC.

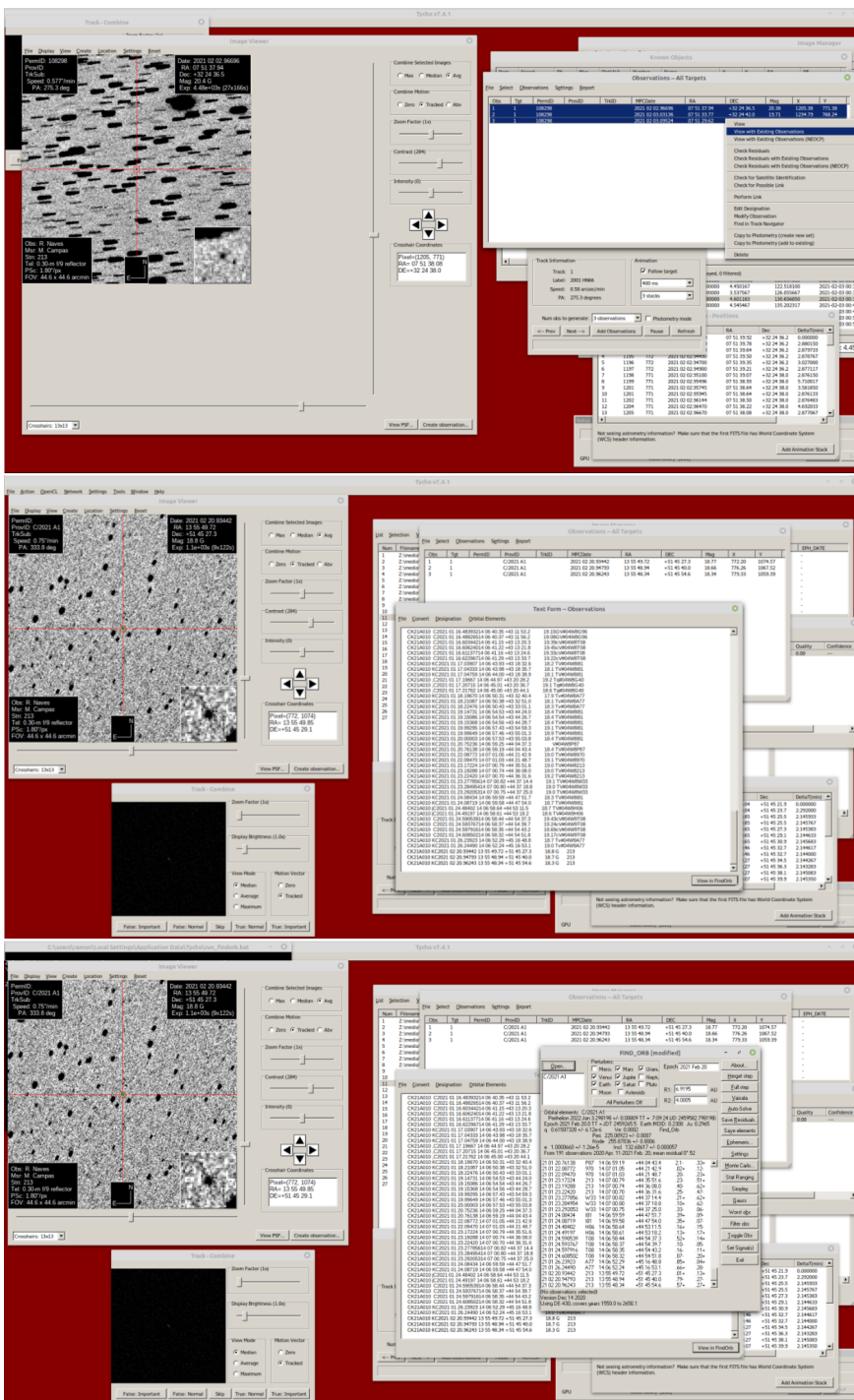


Figura 16. Procedimiento para verificar la calidad de las observaciones.

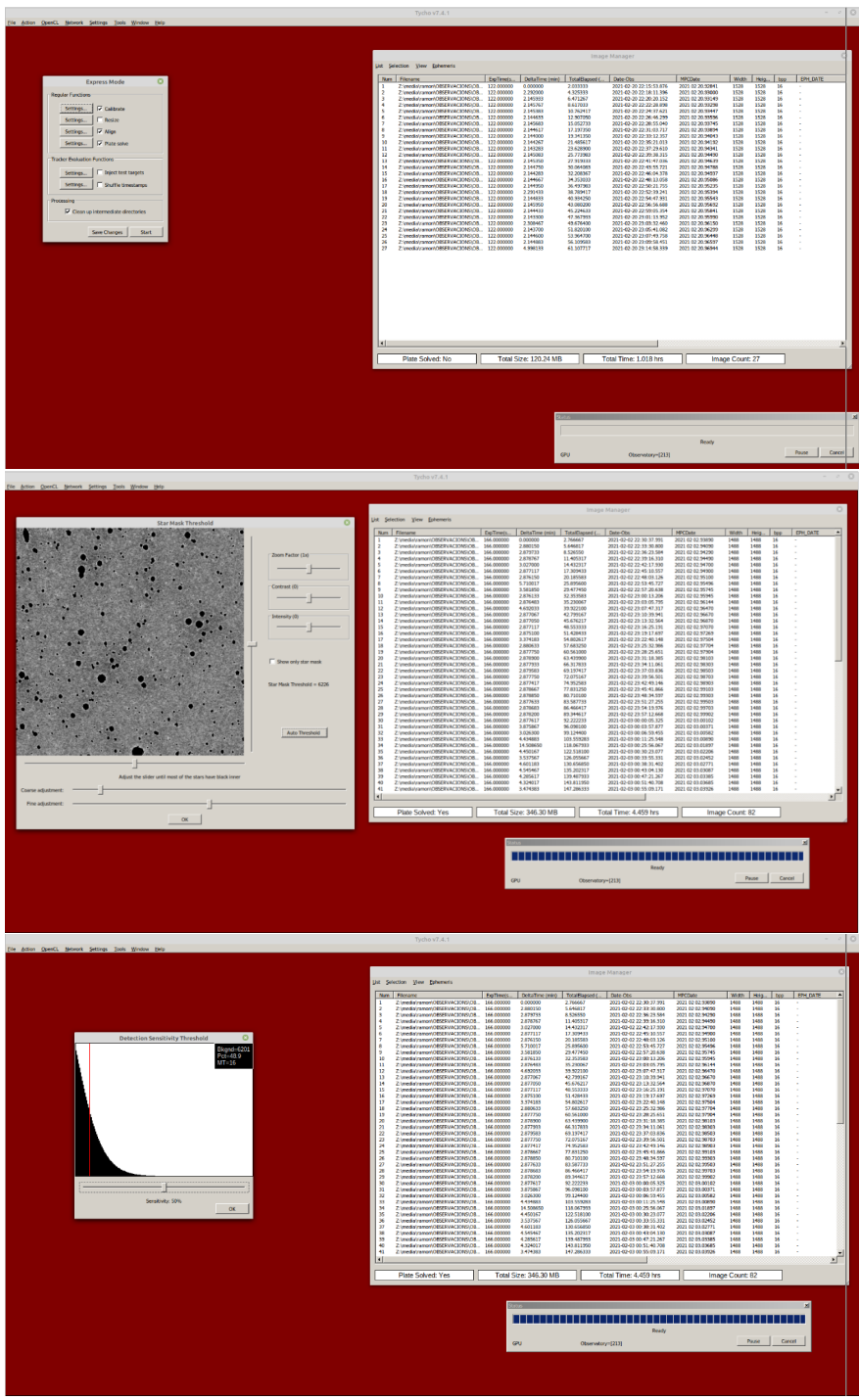


Figura 17. Procedimientos para operar con la herramienta Synthetic tracker.

por tanto un objeto real. Una vez verificado el objeto, clicamos dentro de la ventana **Verify Track** el botón **Add observations**, Fig. 14. Nos aparecerá después el siguiente menú para añadir el nombre del cometa o asteroides a medir, en este caso **C/2021 A1**, sin ningún añadido de nombre más. Luego clicaremos en **Ok**. Nos aparecerá el número de observaciones que hayamos realizado (en este caso tres).

Dentro de la ventana **Observations All Targets** podremos generar un informe para el **Minor Planet Center (MPC)** para enviar por correo electrónico (**Generate MPC1992 Report**), o un informe en formato **ADES** para enviar vía **Web (Generate ADES Report)**, Fig. 15.

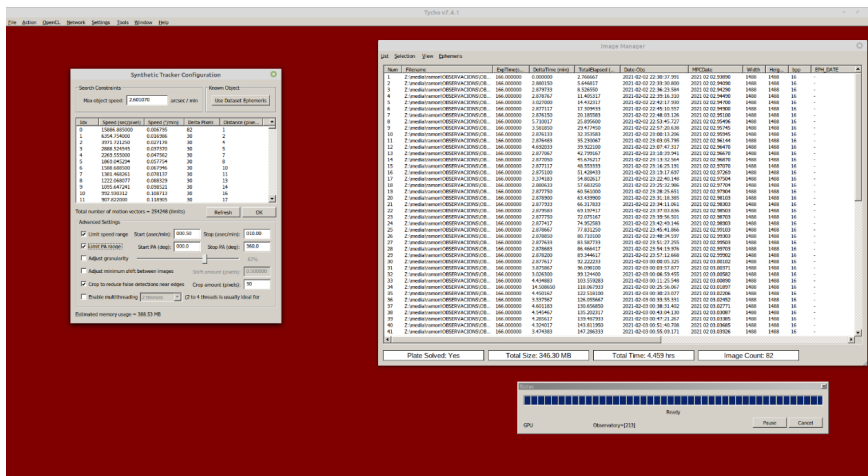


Figura 18. Establecimiento de parámetros para iniciar la herramienta de búsqueda de objetos Synthetic tracker.

6. Comprobación de la calidad de las observaciones astrométricas

Si queremos, podemos verificar la calidad de nuestras observaciones descargándonos más observaciones del MPC y verificándolas posteriormente con Findorb. Para ello (Fig. 16):

1. Clicamos dentro de la ventana **Observations All Targets** en **View with Existing Observations**. Esto nos abrirá una nueva ventana con todas las observaciones disponibles del MPC para este objeto, añadiendo las nuestras al final.
2. Clicamos en esta nueva ventana **Text Form Observation** (abajo, donde aparece **View in Findorb**). Nos aparecerá entonces una nueva ventana con el programa **FindOrb** donde podremos ver los residuos de nuestras observaciones.

7. Synthetic tracker

Esta potente herramienta sirve para buscar objetos desconocidos o perdidos en nuestras imágenes, que no figuran en las bases de datos del MPC. Para ello (Fig. 17):

1. Seleccionamos en **Image Manage** la lista de imágenes, preanalizadas, donde queramos buscar o intentar descubrir nuevos cometas o asteroides.
2. En el menú **Actions** de la barra principal clicamos **Synthetic tracker** (último ítem de ese menú desplegable).

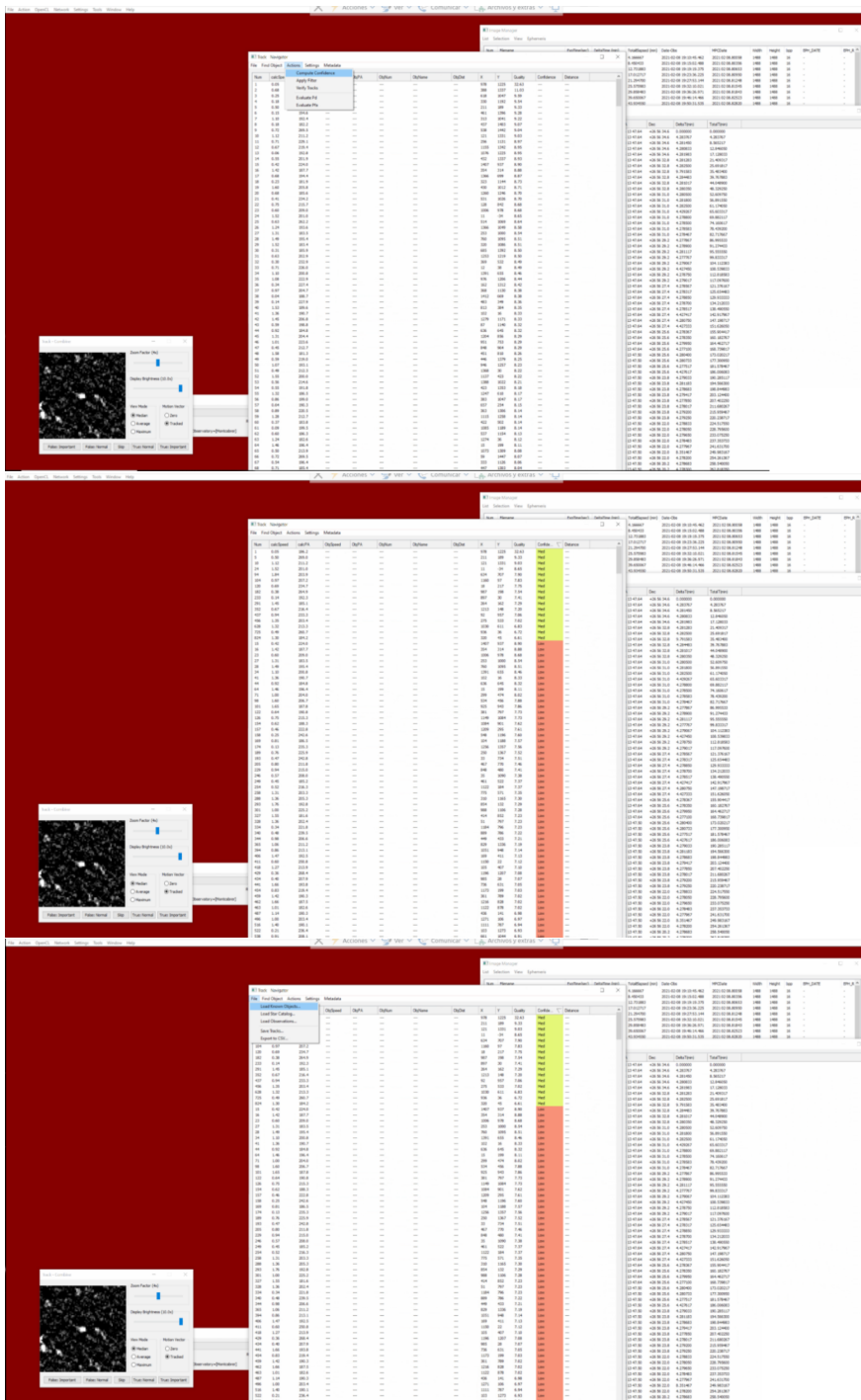


Figura 19. Lista de objetos candidatos y cálculo del nivel de confianza.

Nos aparece una nueva ventana *Star Mask Threshold*, donde simplemente clicamos en *Auto Threshold*. Luego, en la otra ventana que nos aparece, seleccionamos la sensibilidad, aunque el valor por defecto, 50%, parece ir muy bien.

Por último, en la nueva ventana que aparece, Fig. 18, podemos no marcar nada (en este caso el programa tendrá que hacer muchos más cálculos), o podremos limitar la velocidad (*Limit seed range*) y el ángulo (*Limit PA Range*) para restringir el número de cálculos necesario.

Por ejemplo, para buscar asteroides del cinturón principal en oposición, podemos poner velocidades de 0.4 a 0.8, y ángulos entre 250° y 280°. De todas formas, podemos no poner nada para una búsqueda exhaustiva que incluirá todo tipo de objetos: NEOs, Transneptunianos, cometas, etc., sólo que el proceso tardara más tiempo en realizarse.

Para terminar clicamos en *Ok* y esperamos que el proceso concluya; puede tardar desde minutos a muchas horas, y nos hará un pronóstico del tiempo requerido. Podemos abortar el proceso una vez iniciado y transcurrido cierto tiempo. De todos modos el programa nos enseñará los posibles objetos que haya tenido tiempo de encontrar antes de que se abortara el proceso.

Una vez terminado (o abortado) este, se nos mostrará una lista de posibles candidatos. Nos pueden aparecer objetos perdidos o incluso por descubrir, pero tenemos que ser meticulosos, pues muchos de ellos no serán más que artefactos fruto del ruido, rayos cósmicos, etc. En la lista de candidatos *Track Navigator* clicaremos *Compute confidence*, Fig. 19. Así calculará la probabilidad de los mismos, y clicando sobre *Confidence* los ordenará de mayor a menor probabilidad. A continuación podemos clicar en *Load Known Objects* y así nos indicará cuáles de ellos son conocidos. Finalmente, clicando encima de cada uno de ellos se nos realizará un *stack*, y procederemos de la misma forma que siempre. En el caso del ejemplo que hemos mostrado se trata de un asteroide ya conocido.

References

[1] R. Naves y M. Campàs, *JCAAC* **1**, 53 (2024).

1001011101001100
1111001010011110
1001011101001100
1111001010011110
00e9 4819
fdf0 8888 0098 7667
778a 8828 8888 003c
3b82 0010 0000
0004 0218 7b7a
c7c8 089a ffff

```
import math  
def kepler_solver(eccentricity, mean_anomaly, tolerance=1e-10, max_iterations=1000):  
    eccentric_anomaly = mean_anomaly # Initial guess  
    for _ in range(max_iterations):  
        f = eccentric_anomaly - eccentricity * math.sin(eccentric_anomaly) - mean_anomaly  
        f_prime = 1.0 - eccentricity * math.cos(eccentric_anomaly)  
        eccentric_anomaly -= f / f_prime  
    if abs(f) < tolerance:  
        return eccentric_anomaly
```

