



*Facultad  
de  
Ciencias*

**Simulación, detección y caracterización de  
tránsitos de exoplanetas con la misión  
espacial Kepler**

Simulation, detection and characterization of  
exoplanet's transits with Kepler space mission

Trabajo de Fin de Grado  
para acceder al

**GRADO EN FÍSICA**

Autor: Iván Villegas Pérez

Director: Diego Herranz Muñoz

Codirectora: Rosa María  
Domínguez Quintero

Junio 2023



*A mi familia, por apoyarme durante todos estos años e impulsarme a perseguir mis sueños.*

*A mis amigos (desperdigados por toda España y parte del extranjero), por escucharme siempre con atención a pesar de (seguramente) no entender de qué hablaba.*

*A mis tutores, por la paciencia que han tenido conmigo.*

*Finalmente, a Elena, por creer en mi más de lo que yo mismo creo y ser mi punto de apoyo durante estos meses.*



## Resumen

---

En este Trabajo de Fin de Grado se han estudiado técnicas de detección de exoplanetas y distintas técnicas de limpieza de datos obtenidos por el telescopio espacial Kepler, optando por el uso de un filtro de mediana. Se han estudiado los datos provenientes del satélite Kepler y el ruido de los mismos. A partir de esta información se ha desarrollado un programa Python para leer y limpiar los datos, permitiendo caracterizar el exoplaneta estudiado siempre y cuando se conozca la masa, el radio y la temperatura de la estrella.

En primer lugar, para comprobar el funcionamiento del programa empleado se han realizado distintas simulaciones, solucionando problemas y haciendo el código lo más eficiente posible. Una vez se ha confirmado que el programa desarrollado devuelve valores numéricos compatibles con los valores introducidos, se ha probado cómo de efectivo es este método de limpieza de datos, obteniendo la dependencia del error relativo según la variación de distintos parámetros, viendo la mayor dependencia proviene de variar los periodos del tránsito planetario y, en mayor medida, del periodo de la variabilidad estelar.

La principal ventaja de este método es que, indistintamente de la cantidad de ruido, es capaz de obtener con bastante precisión el periodo del tránsito planetario. Por el contrario, depende de una forma excesiva del ajuste de la variabilidad de la estrella a una función seno. Esto lleva a una gran fuente de error, principalmente porque muchas veces la variabilidad de las estrellas no está descrita por una curva de este tipo ni tiene por qué seguir este tipo de oscilación.



## Summary

---

In this Final Degree Project, exoplanet detection techniques and various data cleaning techniques obtained from the Kepler space telescope have been studied, going for the use of a median filter. Data from the Kepler satellite and its noise have been analyzed. Based on this information, a Python program has been developed to read and clean the data, enabling the characterization of the studied exoplanet as long as the mass, radius, and temperature of the star are known.

Firstly, to verify the functionality of the used program, various simulations have been conducted, addressing issues and making the code as efficient as possible. Once it has been confirmed that the developed program returns numerical values consistent with the input values, the effectiveness of this data cleaning method has been tested. The dependence of the relative error has been obtained by varying different parameters, with the highest dependence found when altering the periods of the planetary transit and, to a greater extent, the period of stellar variability.

The main advantage of this method is that, regardless of the amount of noise, it is capable of accurately obtaining the period of the planetary transit. However, it excessively relies on fitting the stellar variability to a sine function. This leads to a significant error source, mainly because the variability of stars is often not described by such a curve, nor does it necessarily follow this type of oscillation.



# Índice

---

<b>Índice</b>	<b>I</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Introducción teórica . . . . .	1
Exoplanetas y métodos de detección . . . . .	1
Órbitas y tránsitos . . . . .	3
La transformada de Fourier . . . . .	5
Ruido en la señal . . . . .	7
Variabilidad de la curva de luz de la estrella . . . . .	8
1.2 Kepler . . . . .	9
1.3 Dificultades que presenta el análisis de datos de Kepler . . . . .	11
El filtro de mediana . . . . .	11
1.4 Objetivos . . . . .	12
<b>2 Metodología</b>	<b>15</b>
2.1 Simulación de los datos . . . . .	15
2.2 Limpieza de datos . . . . .	16
El ruido de la señal . . . . .	16
La variabilidad de la curva de luz de la estrella . . . . .	18
2.3 Efectividad del método . . . . .	18
2.4 Caracterización del exoplaneta . . . . .	20
<b>3 Resultados</b>	<b>23</b>
3.1 Resultados de las simulaciones . . . . .	23
Variación del error relativo . . . . .	26
3.2 Aplicaciones a exoplanetas reales detectados por Kepler . . . . .	29
TrES-2 b . . . . .	29
Kepler-75 b . . . . .	32
3.3 Period04 . . . . .	34
<b>4 Conclusiones</b>	<b>37</b>
<b>Bibliografía</b>	<b>41</b>

<b>Código</b>	<b>45</b>
<b>Índice de Figuras</b>	<b>71</b>
<b>Índice de Tablas</b>	<b>75</b>

# 1

## Introducción

---

Durante la última década, el descubrimiento y caracterización de exoplanetas (planetas que orbitan otras estrellas que no son el Sol) está teniendo un gran relieve científico e impacto social. A día de hoy, existen más de cinco mil exoplanetas confirmados en casi cuatro mil sistemas planetarios, y se cuenta con casi otros diez mil posibles candidatos a la espera de confirmación. Gracias a estos descubrimientos hoy se tiene la certeza de que la existencia de sistemas planetarios, tal vez no muy diferentes al nuestro, es un fenómeno frecuente en nuestro entorno galáctico. Muchos de los descubrimientos de exoplanetas se deben al observatorio espacial Kepler, que recopiló datos de miles de estrellas entre 2009 y 2018. Dichos datos están disponibles de forma pública y muchos de ellos aún no han sido analizados en su totalidad.

### 1.1 Introducción teórica

#### Exoplanetas y métodos de detección

Un exoplaneta (o planeta extrasolar) es aquel que orbita una estrella que no es el Sol. Por norma general, los planetas reflejan parte de la luz de su estrella. Este reflejo suele ser muy tenue en comparación al brillo de su estrella, por lo que es prácticamente imposible (únicamente 67 exoplanetas se han detectado de esta manera) llevar a cabo una detección directa. Las pocas imágenes de exoplanetas existentes son de planetas grandes (mucho más grandes que Júpiter), distantes a su estrella y muy calientes, emitiendo una intensa radiación en el espectro infrarrojo. Por el momento, la gran mayoría de los exoplanetas conocidos han sido detectados mediante métodos indirectos. En la actualidad existen varios métodos indirectos para la detección de exoplanetas. A continuación se detallan los principales métodos.[1]

- Método de las velocidades radiales: está basado en el efecto Doppler, esto es, la relación entre la frecuencia emitida y la frecuencia recibida. Técnicamente, ambos cuerpos orbitan en torno al centro de masas del sistema (el

planeta también ejerce una fuerza gravitacional sobre la estrella), por lo que el movimiento oscilatorio de la estrella pueden detectarse mediante variaciones en las líneas espectrales si la estrella se acerca (corrimiento al azul) o se aleja (corrimiento al rojo) del observador. Este método únicamente es efectivo para exoplanetas muy pesados y muy cercanos a la estrella que orbitan. Este fue uno de los primeros métodos en encontrar exoplanetas, confirmando la existencia de más de mil exoplanetas.[1]

- Método de la astrometría: al igual que en el método de las velocidades radiales, en este método se estudia el movimiento de la estrella en torno a un centro de masas del sistema estrella-planeta. Este movimiento, si es lo suficientemente amplio, provoca un cambio en la posición aparente de la estrella. A día de hoy, este método es increíblemente difícil de llevar a cabo (únicamente dos exoplanetas han sido confirmados). Las estrellas apenas se mueven por planetas pequeños como la Tierra, por eso, al igual que el método de las velocidades radiales, únicamente es útil para detectar grandes exoplanetas muy cercanos a su estrella. El uso de este método se basa en tomar una serie de fotografías de un grupo de estrellas y medir las distancias entre la estrella en la que se buscan exoplanetas y las de referencia. En caso de detectar un cambio en estas distancias, se analizaría el cambio en busca de alguna prueba de la existencia de exoplanetas. Este método requiere de sistemas ópticos extremadamente precisos embarcados en satélites, puesto que la atmósfera distorsiona y refracta la luz.[1]
- Método de las microlentes gravitacionales (*microlensing*): se aprovecha de las desviaciones que sufre la luz de una estrella cuando un cuerpo masivo se encuentra entre una estrella y el observador. Esto se traduce en que la estrella emisora se ve más brillante que cuando no está este cuerpo masivo. Durante el transcurso de aproximadamente un mes, la luz de la estrella emisora aumenta en intensidad y luego disminuye. Si existe un exoplaneta lo suficientemente masivo como para curvar la luz significativamente, en este aumento de brillo se observa una irregularidad (figura 1.1).

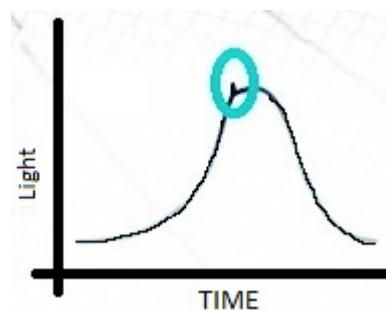


Figura 1.1: Curva de luz observada mientras un sistema masivo se interpone ente el observador y la estrella estudiada. Se aprecia como la intensidad de la luz aumenta y disminuye debido al efecto de *microlensing*. Marcado en azul se ve la irregularidad que produce el exoplaneta al focalizar la luz de la estrella ([5 ways to find a planet](#)).[1]

Las principales desventajas de este método se encuentran en la necesidad de que el planeta sea lo suficientemente masivo y que estos eventos de *microlensing* no se pueden predecir, por lo que es necesario observar una gran región del cielo durante un largo periodo de tiempo. Otra posibilidad a

tener en cuenta es que un planeta errante (un planeta que no orbita ninguna estrella) se cruce en el camino de la luz y cause un efecto de *microlensing*. El principal obstáculo para confirmar estos planetas errantes es que no hay ninguna prueba de que el efecto se deba a un exoplaneta y no a una enana marrón<sup>1</sup> o un agujero negro primordial<sup>2</sup>.<sup>[1]</sup>

- Método de los tránsitos: consiste en el estudio de disminuciones periódicas en la intensidad de la luz de una estrella. Esas disminuciones se deben al paso de un astro entre la estrella observada y el observador. Al observar varias disminuciones periódicas en la intensidad lumínica de una estrella se puede obtener un periodo orbital para el cuerpo que orbita dicha estrella. La profundidad y duración de los tránsitos pueden dar mucha información sobre el exoplaneta que lo provoca. Cuanto más profundo sea el tránsito, mayor será el radio del exoplaneta, y cuanto más dure el tránsito, mayor será la distancia entre la estrella y el exoplaneta. Dada la naturaleza del método existe un fuerte sesgo, favoreciendo la detección de exoplanetas de mayor tamaño (puesto que la disminución en la intensidad será mayor). Por otro lado, la gran ventaja de este método es que la distancia entre el exoplaneta y la estrella no es relevante.<sup>[1]</sup>

Una vez expuestas las distintas formas de proceder, se ha optado por trabajar con el método de los tránsitos por lo intuitivo que es su funcionamiento y por no necesitar de planetas extremadamente masivos y/o cercanos a su estrella. Como se ha expuesto con anterioridad, con este método no solo se pueden hallar exoplanetas, sino que también se pueden caracterizar. Además, ha sido y es el método más eficaz, con más de 4000 exoplanetas confirmados.

## Órbitas y tránsitos

Hasta ahora se ha hablado del movimiento de planetas alrededor de estrellas (o más bien del movimiento de ambos alrededor del punto de centro de masas). Este movimiento sigue una trayectoria que se denomina “órbita”. Estas órbitas forman una elipse alrededor del centro de masas.

Esta elipse puede acercarse más, o menos, a una circunferencia según la excentricidad de la órbita. La excentricidad es un parámetro que se utiliza para cuantificar la desviación de una órbita de una circunferencia perfecta. La excentricidad ( $e$ ) viene dada por la ecuación 1.1.

$$e = \frac{r_a - r_p}{r_a + r_p} = 1 - \frac{2}{\frac{r_a}{r_p} + 1}, \quad (1.1)$$

donde  $r_a$  es el radio en el apoápside (la distancia más lejana de la órbita al centro de masas) y  $r_p$  es el radio en el periápside (la distancia más cercana de la órbita al centro de masas).

<sup>1</sup>Estrellas poco masivas y relativamente frías, entre los 1000K y los 1200K

<sup>2</sup>Agujeros negros teorizados formados por la extrema densidad del Universo al inicio de su expansión. Estos agujeros negros primordiales tendrían una masa varios órdenes de magnitud menores que las del Sol, por lo que podrían parecer planetas errantes.

Así, si la órbita describe una circunferencia perfecta ( $r_p = r_a$ ), la excentricidad tiene un valor de  $e = 0$ , mientras que si la órbita describe una elipse, cuanto más plana sea esta (cuanto más se diferencie de una circunferencia perfecta), mayor será el valor de la excentricidad estará entre cero y uno ( $0 < e < 1$ ). En caso de que la excentricidad sea mayor o igual que uno ( $e \geq 1$ ) ya no se podría hablar de órbita, sino de trayectoria parabólica ( $e = 1$ ) o hiperbólica ( $e < 1$ ).

Dado que las órbitas son trayectorias cerradas alrededor de un punto, es un movimiento periódico, es decir, se repite en el tiempo. El tiempo que tarda un cuerpo en recorrer esta órbita es lo que se llama periodo orbital. Dentro del periodo orbital, se ha estudiado el tránsito, que es el paso de un planeta por delante de su estrella, lo que conlleva una disminución de su brillo. La profundidad de este tránsito dependerá del tamaño del planeta.

Otro parámetro a la hora de determinar una órbita es la inclinación de la misma ( $i$ ). Esto es, el ángulo que forma el plano que contiene la órbita respecto al plano orbital de referencia; el plano que contiene la órbita de la Tierra. Este, junto a la excentricidad y el semieje mayor, es uno de los parámetros básicos para determinar una órbita, ya que define el plano que la contiene.

Ahora que el concepto de órbita ya ha sido explicado, se pasará a explicar los tránsitos. Cómo se ha mencionado anteriormente, un tránsito es el paso de un astro por delante de otro, provocando una disminución en la intensidad del brillo. Un pequeño esquema puede verse en la figura 1.2<sup>3 4</sup>.

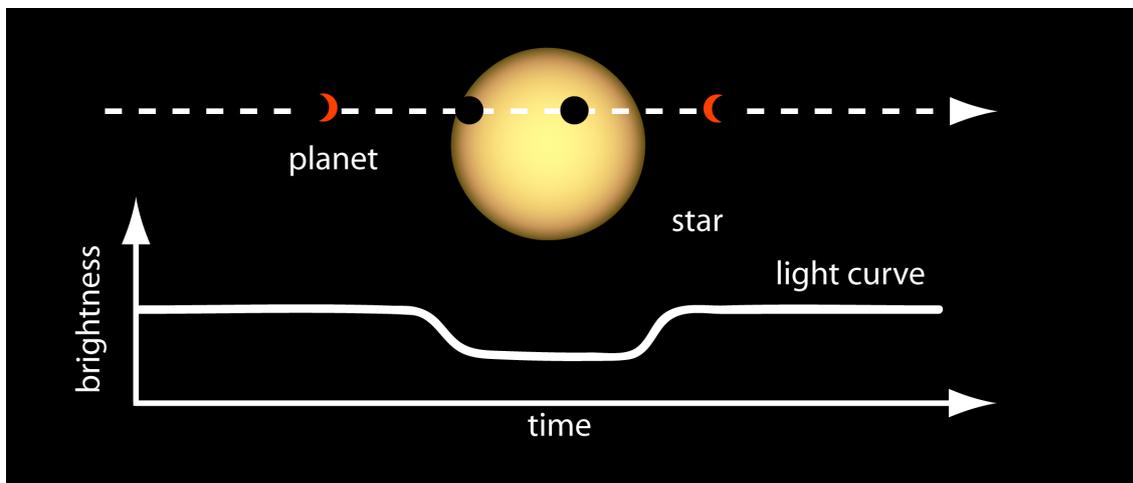


Figura 1.2: Imagen que muestra el concepto de “tránsito”. Se puede apreciar que según el planeta atraviesa la superficie de la estrella, el brillo de la misma disminuye. Cuantizando el descenso, se puede determinar el ratio entre el radio del planeta y el radio de la estrella (*5 ways to find a planet*).[1]

En este trabajo se ha asumido que el tránsito es parabólico, lo cual implica que el planeta se interpone entre la estrella y el observador de manera gradual, al igual que hace en el momento en el que desaparece de la superficie visible de la

<sup>3</sup>NASA Ames.

<sup>4</sup>[https://www.nasa.gov/mission\\_pages/kepler/multimedia/images/transit-light-curve.html](https://www.nasa.gov/mission_pages/kepler/multimedia/images/transit-light-curve.html)

estrella.

La curva de luz de una estrella con tránsitos parabólicos,  $B(t, P, t_0, d, p)$ , está definido por la ecuación 1.2.

$$B(t, P, t_0, d, p) = -\text{máx} \left( 0, p \left[ 1 - \frac{4}{d^2} \left( t - t_0 - \left[ \frac{t - t_0 + P/2}{P} \right] P \right)^2 \right] \right), \quad (1.2)$$

donde  $t$  es el tiempo,  $P$  el periodo orbital del planeta,  $t_0$  es el tiempo del centro de cualquier tránsito,  $d$  es la duración del tránsito y  $p$  es la profundidad (cuánto disminuye).

### La transformada de Fourier

Una vez está claro el método a seguir, la pregunta que surge de forma natural es ¿cómo se puede determinar el periodo orbital del exoplaneta a partir de las disminuciones de intensidad de luz de la estrella? A continuación se explican algunos de los métodos más usados.

- Método de autocorrelación: está basa en la medida de la correlación entre un conjunto de datos y una versión retardada de sí mismo. Al calcular la autocorrelación de este conjunto de datos para diferentes desplazamientos del conjunto es posible identificar periodos repetitivos, indicados por los picos en la función de autocorrelación,  $R(j)$ .

$$R(j) = \sum_n (x_n - m)(x_{n-j} - m), \quad (1.3)$$

donde  $j$  es el dato en el que se calcula la correlación,  $x_n$  es cada dato del conjunto de datos y  $m$  es el valor esperado de  $x_n$ .

- Método de las medias móviles: está basado en el cálculo de la media de los datos en intervalos de tiempo consecutivos. Observando los promedios en diferentes intervalos es posible identificar patrones repetitivos y posibles periodos en una serie de datos.
- La Transformada de Fourier: es una herramienta matemática que permite descomponer una señal en sus componentes de frecuencia. Al aplicar la transformada de Fourier a un conjunto de datos, se obtiene una representación en el dominio de frecuencias. Los picos en el espectro de potencias indican las frecuencias dominantes en el conjunto de datos y, por tanto, los posibles periodos presentes.[2][3]

La transformada de Fourier viene dada por la ecuación 1.4.

$$\hat{F}(f) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(t) e^{-ift} dx, \quad (1.4)$$

donde  $\hat{F}$  es la transformada de Fourier de  $F$ , que es la función que define la señal con la que se trabaja. Por otro lado,  $t$  es el tiempo del dominio temporal de la señal y  $f$  es la frecuencia del dominio de frecuencias.

El espectro de potencias se define como la transformada de Fourier de la función de autocorrelación (ecuación 1.3). El espectro de potencias permite determinar la distribución de la potencia de una señal a lo largo de un intervalo de frecuencias.[4]

- Otra forma de determinar el periodo es intentar ajustar los datos a un modelo matemático y utilizar el mismo para predecir futuros valores. Observando los patrones de las predicciones es posible identificar los posibles periodos presentes en el conjunto de datos disponible.

Dado que no se puede asumir un único modelo matemático que se ajuste a todos los tránsitos, este último método queda descartado. Los dos primeros métodos no parece que vayan a alcanzar la precisión que se pretende obtener en la determinación del periodo, ya que parecen métodos relativamente simples. Por estos motivos, se ha optado por usar la transformada de Fourier como método para determinar el periodo orbital de los exoplanetas.

Es bien sabido que, para señales largas, la implementación directa de la transformada de Fourier (FT, por sus siglas en inglés; *Fourier Transform*) puede ser increíblemente costosa. Es en este momento cuando entra en juego la transformada rápida de Fourier (FFT, por sus siglas en inglés; *Fast Fourier Transform*), un algoritmo eficiente para calcular la FT cuando la señal de entrada tiene una longitud que es potencia de dos. La FFT realiza  $N\log(N)$  ( $N$  es la longitud de la señal) operaciones frente a las  $N^2$  que realiza la FT, reduciendo considerablemente el tiempo de cálculo. Esto hace que la FFT sea mucho más rápida y práctica para su implementación en programas.[2][3]

La FFT viene dada por la ecuación 1.5.

$$\tilde{F}_k = \sum_{n=0}^{N-1} F_n e^{-ikn\left(\frac{2\pi}{N}\right)}, \quad k = 0, \dots, N-1, \quad (1.5)$$

donde  $\tilde{F}$  es la transformada discreta de la señal,  $N$  es el periodo de la señal y  $F_n$  es cada dato de la señal a transformar.

Una vez obtenida la transformada de Fourier de la señal, cada máximo de los valores calculados es una frecuencia de repetición de los datos. El valor máximo indica la frecuencia dominante ( $f_{\text{dom}}$ ), en este caso concreto, esta frecuencia debería ser la frecuencia con la que se produce un tránsito planetario. Por tanto, el periodo orbital del exoplaneta ( $P$ ) estudiado será la inversa de esa frecuencia dominante (ecuación 1.6).

$$P = \frac{1}{f_{\text{dom}}}. \quad (1.6)$$

Dado que la FFT calcula la frecuencia de repetición de los datos, no solo se obtienen las frecuencias menores, sino que también se obtienen todos sus armónicos (múltiplos). Esto se debe a que el término  $e^{-ikn(\frac{2\pi}{N})}$  de la ecuación 1.5 es periódico y cualquier múltiplo de una frecuencia concreta va a tener el mismo resultado. En otras palabras, si cierto dato se repite cada 2 días, también se repite cada 4 días<sup>5</sup>. Según aumenta esta frecuencia de repetición se repite menos veces el dato; por decirlo de alguna manera, si un dato se repite cada 2 días, si se busca una repetición cada 4 días, el número de veces que se va a repetir va a ser menor (la mitad de las veces). Este es el motivo de elegir la frecuencia del máximo como frecuencia dominante.

No obstante, pueden aparecer picos adicionales debidos otras frecuencias que no sean la frecuencia dominante y sus armónicos. Estos picos secundarios pueden estar causados por ruido, interferencias de la señal, o por características propias de la señal. Este es el motivo principal por el que se deben limpiar los datos. Así, aislando la señal puramente debida al tránsito, se puede determinar con mucha precisión el periodo orbital.

### Ruido en la señal

La señal proporcionada por Kepler tiene una fuerte sistemática instrumental. En 2020, Jakob Robnik y Uroš Seljak determinaron que el ruido de la señal se tenía una componente gaussiana y otra componente debida a un ruido no gaussiano que sigue una distribución  $t$  no central (NCT, por sus siglas en inglés; *Non-Central  $t$  distribution*).[5][6]

Esta parte de ruido no gaussiano proviene de errores instrumentales sistemáticos como cambios de temperatura en la apertura o a la órbita que sigue el satélite, teniendo en cuenta la rotación de  $90^\circ$  que sufre cada tres meses (Sección 1.3).

La función de densidad de esta distribución,  $f_{\text{NCT}}(x|\nu, \delta, \sigma)$ , viene dada por la ecuación 1.7.[7]

$$f_{\text{NCT}}(x|\nu, \delta, \sigma) = \frac{\nu^{\frac{\nu}{2}} e^{-\frac{\sigma^2}{2}}}{\Gamma\left(\frac{\nu}{2}\right) \sqrt{\pi} (\nu + x^2)^{\frac{\nu+1}{2}}} \sum_{r=0}^{\infty} \frac{(x \cdot \delta)^r}{r!} \left(\frac{2}{\nu + x^2}\right)^{\frac{r}{2}} \Gamma\left(\frac{\nu + r + 1}{2}\right), \quad (1.7)$$

donde  $\nu$  indica los grados de libertad<sup>6</sup>,  $\delta$  es el parámetro no central de la distribución<sup>7</sup>,  $\sigma$  es la desviación estándar y  $\Gamma(z)$  es la función gamma:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt.$$

<sup>5</sup>Y cada  $-4$ . La transformada de Fourier es simétrica respecto al origen, pero dado que no tiene sentido hablar de un periodo orbital negativo, esta parte de la solución puede ser obviada.

<sup>6</sup>Los grados de libertad de una distribución indican cuantos valores pueden ser asignados de manera aleatoria, antes de que el resto de datos tomen un valor automáticamente.

<sup>7</sup>Desviación de la media respecto al cero.

Los grados de libertad de una distribución NCT están relacionados con el tamaño de la muestra, afectando la forma de la distribución. Cuanto mayor sea el número de grados de libertad, más se asemejará la distribución NCT a la distribución normal. Esto implica que, cuanto menor sea este valor, mayor peso estadístico tendrán los valores más extremos.[8]

En la figura 1.3 se muestra lo ruidosas que pueden llegar a ser las curvas de luz y a consecuencia de ello, la necesidad de limpiar los datos con la esperanza de poder detectar algún exoplaneta.

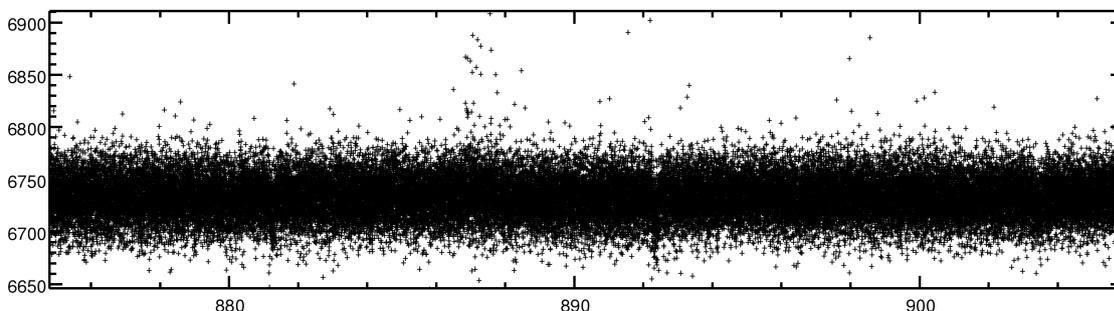


Figura 1.3: Curva de luz de la estrella Kepler-228. Se observa claramente que es una curva de luz extremadamente ruidosa, lo cual implica una gran necesidad de limpiar los datos. Debido al ruido de la señal recibida, en este caso no se pueden apreciar los tránsitos, al menos a simple vista.

### Variabilidad de la curva de luz de la estrella

En general, las curvas de luz de las estrellas estudiadas por la misión espacial Kepler no son variables. No obstante, pudiera darse el caso de que observe un sistema de estrellas binarias eclipsantes (un sistema de dos estrellas cuyo plano orbital está orientado hacia la Tierra y sufren eclipses y tránsitos mutuos) o cefeidas<sup>8</sup> (estrella que pulsa radialmente variando su tamaño como su temperatura, produciendo cambios de brillo con amplitud y periodo muy regulares), en cuyo caso se obtiene que además de los tránsitos, la curva de luz presenta una variabilidad en los momentos en los que no se produce tránsito alguno. En la figura 1.4 se ve un ejemplo de curva de luz oscilante debido a un sistema de binarias eclipsantes (dos estrellas) denominado “Kepler-47”. [9]

Como generalización y con intención de poder aplicar el programa desarrollado (disponible en el Capítulo 4 y en el repositorio público de GitHub “[Simulation-detection-and-characterization-of-exoplanet-transits-with-Kepler-space-mission](https://github.com/ivanvillegas7/Simulation-detection-and-characterization-of-exoplanet-transits-with-Kepler-space-mission)”<sup>9</sup>) al mayor número posible de estrellas con tránsitos planetarios, se ha asumido que todas las curvas de luz tienen una variabilidad periódica. No tener variabilidad en la curva de luz es un caso particular en el que la amplitud de la variabilidad es nula.

<sup>8</sup>Las cefeidas no son el único tipo de estrella variable. Adicionalmente, no es algo que se espere encontrar, puesto que las estrellas que observó Kepler eran estrellas de la secuencia principal.

<sup>9</sup><https://github.com/ivanvillegas7/Simulation-detection-and-characterization-of-exoplanet-transits-with-Kepler-space-mission>

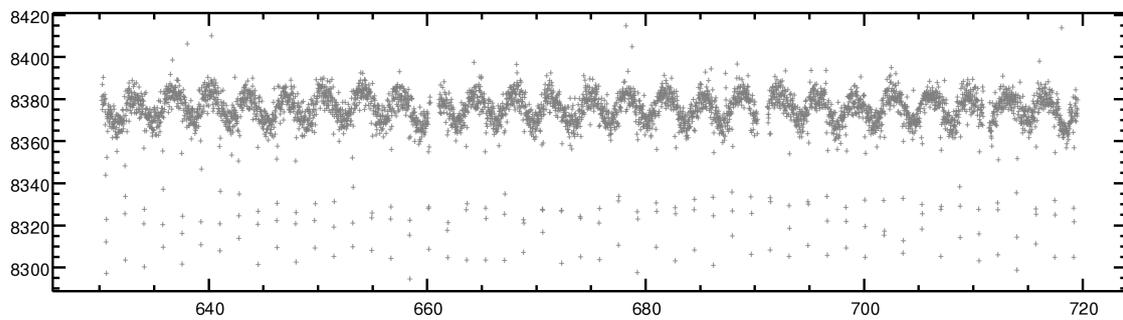


Figura 1.4: Curva de luz de la estrella Kepler-47. Se aprecia claramente que la mayoría de puntos forman una curva sinusoidal fuera del tránsito debido a un sistema de binarias eclipsantes (Orosz et al., 2019).[9].

## 1.2 Kepler

La misión espacial Kepler (el satélite puede verse en la figura 1.5<sup>10 11</sup>) recopiló datos de estrellas en nuestra zona de la Vía Láctea, en concreto en la región de las constelaciones *Lyra* y *Cygnus*, con el objetivo de detectar y caracterizar exoplanetas a partir de los tránsitos de estos en las curvas de luz de las estrellas. El motivo de apuntar a un solo grupo de estrellas durante toda la misión es el gran aumento de la estabilidad fotométrica y la simplificación del satélite (y, por tanto, su presupuesto).

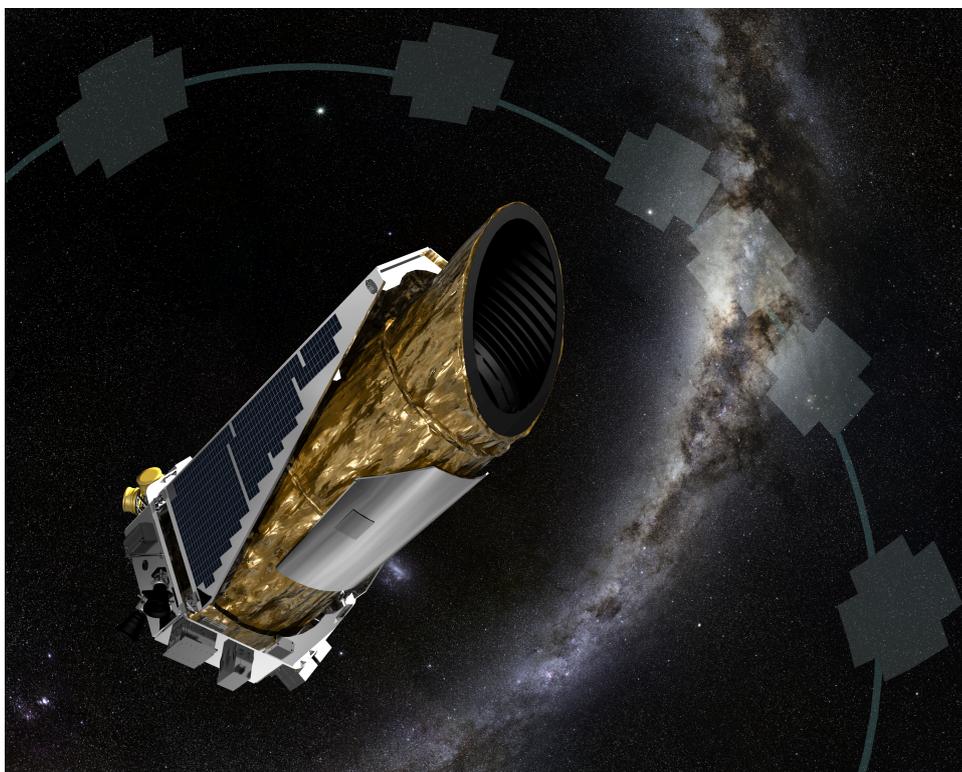


Figura 1.5: Representación artística del satélite Kepler (*Reborn Kepler Can Still Find Planets*).

El fotómetro a bordo de Kepler era un instrumento con un único objetivo: anotar

<sup>10</sup>NASA Ames/JPL-Caltech/T Pyle.

<sup>11</sup><https://www.nasa.gov/ames/kepler/reborn-kepler-can-still-find-planets>

los datos de un único grupo de estrellas durante la duración de la misión. El fotómetro estaba acoplado a un telescopio Schmidt con una apertura inferior a un metro (0.95m) y un campo de visión (FoV por sus siglas en inglés; *Fild of View*) de 105 grados cuadrados (un diámetro de aproximadamente  $12^\circ$ ).

El satélite estaba constituido por un único instrumento, un fotómetro con un *array* de 42 CCDs (figura 1.6<sup>12 13</sup>) en el plano focal, cada uno con más de 2.5 millones de píxeles. Los CCDs se leían cada tres segundos para evitar la saturación de los mismos. Solo se guardaban los datos de estrellas con una magnitud superior a 16 en la frecuencia o longitud de onda correspondiente al color rojo ( $\lambda_{\text{rojo}} = 674\text{nm}$ ).



Figura 1.6: Plano focal del satélite con los 42 CCDs colocados en 21 de los 25 módulos se empleaban para realizar observaciones. Los otros cuatro módulos se utilizaban para guiar el satélite (*Kepler focal plane assembly*).

El fotómetro tenía la sensibilidad suficiente como para detectar tránsitos de planetas del tamaño a la Tierra orbitando una estrella G2V (estrella de la secuencia principal<sup>14</sup> con una masa y temperatura similar al Sol;  $M = 1.07M_{\odot}$ <sup>15 16</sup> y  $T = 5800\text{K}$ ) con magnitud<sup>17</sup> aparente visual de 12 con un intervalo de confianza

<sup>12</sup>NASA/Kepler mission

<sup>13</sup><https://www.nasa.gov/image-feature/kepler-focal-plane-assembly>

<sup>14</sup>El consumo de hidrógeno es la principal fuente de la luminosidad estelar.

<sup>15</sup> $1M_{\odot} = 2 \cdot 10^{30}\text{kg}$

<sup>16</sup>El símbolo  $\odot$  significa solar, por lo que  $M_{\odot}$  es la masa del Sol.

<sup>17</sup>Cantidad que cuantifica el brillo de una estrella

de  $4\sigma$ , con un tiempo de exposición de 6.5 horas[10]. El fotómetro contaba con un filtro que únicamente permitía el paso de fotones con longitud de onda comprendida entre los 400 y los 900nm.

Los datos recopilados por la misión espacial Kepler pueden encontrarse en su catálogo *Kepler Objects of Interest*<sup>18</sup>. Se puede filtrar la búsqueda por parámetros estelares, del exoplaneta o por nombre. En este trabajo se han utilizado los datos de los exoplanetas TrES-2 b (Kepler-1 b) y Kepler-75 b. De esta página se han sacado también los datos estelares necesarios (masa, radio y temperatura) para la caracterización de estos exoplanetas.

### 1.3 Dificultades que presenta el análisis de datos de Kepler

Los datos recopilados por Kepler pueden dividirse en dos grupos, los de cadencia corta y cadencia larga. Estos últimos presentan un mayor tiempo de exposición y, por tanto, una cantidad de puntos insuficientes para un buen análisis. Dado que los datos de cadencia corta han tenido un tiempo de exposición menor, hay una mayor cantidad de datos en un mismo periodo de tiempo. Por este motivo, se ha decidido trabajar siempre con los datos de menor tiempo de exposición.

Por otro lado, el satélite Kepler seguía una órbita heliocéntrica y cada tres meses debía rotar  $90^\circ$  para seguir apuntando al mismo grupo de estrellas. Por este motivo, cada fichero de datos tiene una duración de tres meses. Esto supone un problema porque pudiera darse el caso de que haya planetas con periodos orbitales mayores a estos tres meses, no quedando registrados en algunos de estos paquetes de datos.

#### El filtro de mediana

Dado que la señal ha tenido un tiempo de exposición bajo, esta es extremadamente ruidosa. Ha sido necesario aplicarle un filtro para así poder trabajar con una señal más limpia y clara. Entre los muchos filtros existentes, se ha optado por el uso del filtro de mediana, tanto por su sencillez como por su potencia. El filtro de mediana, al igual que el filtro de la media, es un filtro de paso bajo. El objetivo de este tipo de filtros es suavizar los datos, siendo especialmente útiles en conjuntos de datos muy ruidosos. Este filtro es especialmente útil para eliminar *outliers* (valores atípicos) de la señal. La principal ventaja del filtro de mediana es que el valor resultante es un dato real (la mediana de los datos), reduciendo efectos borrosos y siendo menos sensible que otros filtros de paso bajo a valores extremos. Esto resultará de especial importancia al tener datos con ruido no gaussiano.

El funcionamiento del filtro es sencillo. Se determina una cantidad de puntos a considerar (ventana) y, para cada punto, se tiene en consideración una cierta

<sup>18</sup><https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative>

cantidad de puntos a derecha e izquierda de los datos originales, la mitad del tamaño de la ventana por cada lado. Una vez seleccionados los puntos a emplear, los datos son ordenados y se selecciona la mediana (el dato que quede en mitad de la serie ordenada), reemplazando el punto en el que se filtra la señal por este valor. Por este motivo, los *outliers* no afectan especialmente el resultado. Por otro lado, cuando el punto que se quiere filtrar no tiene suficientes datos a la derecha o a la izquierda, la ventana se completa con ceros, quedando los datos distorsionados en los extremos. En la figura 1.7 puede verse un esquema de su uso a la hora de filtrar imágenes.[11]

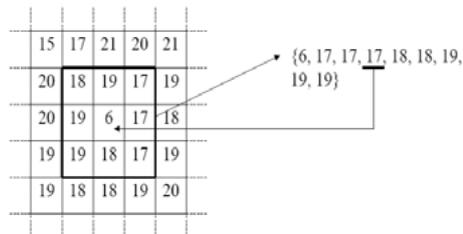


Figura 1.7: Esquema del funcionamiento del filtro de mediana para el filtrado de imágenes. Para filtrar datos el proceso es el mismo, pero en lugar de trabajar con matrices de datos se hace con vectores (Chávez-Burbano, Sarmiento y Ávila, 2009).[11]

Por otro lado, este filtro también tiene sus desventajas. La principal es la pérdida de detalles sutiles. En nuestro caso, esto se traduce en una menor profundidad del posible tránsito tras la aplicación del filtro. Otro inconveniente es la complejidad computacional, puesto que al tener que ordenar los datos y tomar la mediana se emplea mucho más tiempo que en otros filtros, como el filtro de la media.

## 1.4 Objetivos

El principal objetivo de este Trabajo de Fin de Grado ha sido el estudio y caracterización de exoplanetas a partir de la determinación de su periodo orbital.

Para ello se han aplicado técnicas de análisis de datos estelares para el estudio de las curvas de luz de estrellas observadas con la misión espacial Kepler. Debido a la variabilidad intrínseca de las curvas de luz de algunas estrellas y a la presencia de sistemática instrumental fuertemente no gaussiana, las curvas de luz de Kepler son muy ruidosas.

En esta investigación se han aplicado técnicas de reducción de ruido para optimizar la detectabilidad de exoplanetas y la estimación de sus parámetros físicos (periodo, distancia a la estrella, tamaño, inclinación y temperatura promedio).

A modo de resumen, se presentan los objetivos en la siguiente lista:

- Simular una curva de luz.
- Limpiar los datos.

- Caracterizar el exoplaneta.
  - Aplicar estas técnicas a datos reales observados por la misión espacial Kepler, en concreto a los exoplanetas TrES-2 b y Kepler-75 b.
  - Determinación del periodo.
  - Determinación de la distancia a la estrella.
  - Determinación del radio del exoplaneta.
  - Determinación de la inclinación de la órbita.
  - Determinación de la temperatura promedio del exoplaneta.



# 2

## Metodología

### 2.1 Simulación de los datos

En una primera fase, se ha desarrollado un código para simular de forma realista datos como los obtenidos por el satélite Kepler, incluyendo en dichas simulaciones la curva de luz intrínseca estelar, con tránsitos de exoplanetas, mediante la implementación de la ecuación 1.2 (*listing 4*, figura 2.1), y variabilidad sinusoidal (*listing 3*), junto a ruido instrumental no gaussiano (el código utilizado para generar el ruido no gaussiano fue desarrollado por Robnik y Seljak en [5]; *listing 2*). El resultado se muestra en la figura 2.2.

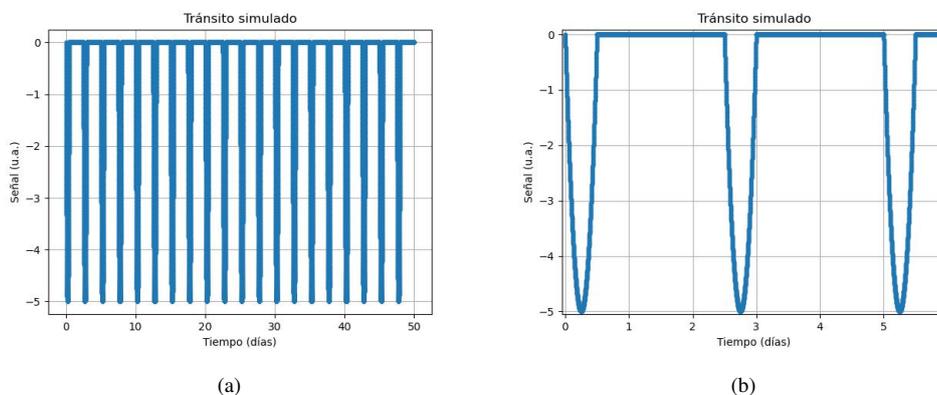


Figura 2.1: Tránsito puro simulado. Se observan disminuciones periódicas de la señal a lo largo de los 50 días simulados en 2.1(a). En 2.1(b) se puede observar los tránsitos con mayor claridad.

Dichas simulaciones han servido para estudiar la efectividad de las técnicas de reducción y análisis de señal que se pretenden utilizar. Una vez determinada la capacidad de dichas técnicas y sus limitaciones, se han aplicado a datos reales obtenidos por la misión Kepler, estudiando sistemas bien conocidos. Concretamente, se han estudiado los exoplanetas TrES-2 b y Kepler-75 b.

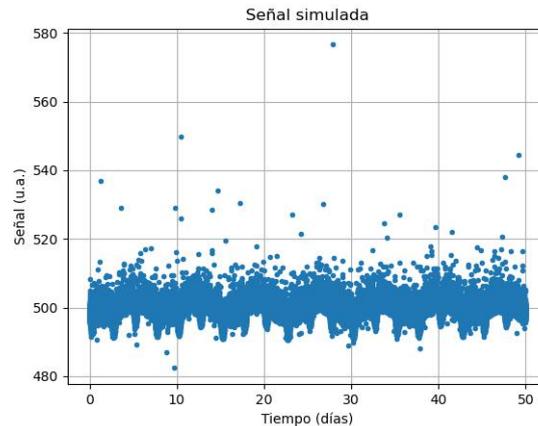


Figura 2.2: Datos simulados. Resultado obtenido al sumar el tránsito puro junto a la variabilidad de la curva de luz debida a una función seno, el *offset* y el ruido.

## 2.2 Limpieza de datos

Dada la importante sistemática instrumental y la posible variabilidad de la curva de luz, en esta investigación se han realizado dos tipos de limpieza; primero eliminar el ruido de la señal y después eliminar la posible variabilidad debida a la estrella.

La importancia de realizar una buena limpieza de los datos radica en la necesidad de aislar el tránsito para que esta repetición periódica se sobreponga a las demás. Entonces, a la hora de obtener la frecuencia dominante mediante la transformada de Fourier, se puede obtener la frecuencia del tránsito y, por ende, el periodo orbital ( $P$ ) del exoplaneta estudiado.

### El ruido de la señal

La naturaleza no gaussiana del ruido de este tipo de señales se debe a que el ruido es una mezcla entre ruido gaussiano y ruido no gaussiano debido a la electrónica y tipo de órbita del satélite, que sigue una distribución  $t$  no central, de forma que la función de probabilidad del ruido,  $p(x, A, \mu, \sigma, \nu, \delta)$ , se define según la ecuación 2.1.[5]

$$p(x, A, \mu, \sigma, \nu, \delta) = (1 - A) \cdot f_{\text{ND}}(\mu, \sigma) + A \cdot f_{\text{NCT}}\left(\frac{x - \mu}{\sigma}, \nu, \delta\right), \quad (2.1)$$

donde  $A$  es la fracción de *outliers* (valores atípicos),  $f_{\text{ND}}(\mu, \sigma)$  es la distribución normal (el subíndice “ND” procede de sus siglas en inglés; *Normal Distribution*), con valor medio  $\mu$  y desviación estándar  $\sigma$  y con  $f_{\text{NCT}}\left(\frac{x - \mu}{\sigma}, \nu, \delta\right)$  siendo la distribución  $t$  no central con parámetros  $\nu$  y  $\delta$ , siendo su parámetro de localidad (valor medio de la distribución) y de escala respectivamente. Para normalizar la distribución no central,  $f_{\text{NCT}}$ , a una  $\sigma$  se normaliza la variable  $x$ , tal que

$$\frac{x - \mu}{\sigma}.$$

Como se ha visto en el Capítulo 1, según la estrella observada, los datos pueden ser extremadamente ruidosos (figura 1.3) y la necesidad de limpiar los datos es acuciante.

El procedimiento seguido por los científicos responsables de la misión espacial Kepler comienza con la corrección de la imagen obtenida por Kepler, mejorando así la calidad de la imagen. El proceso continúa con la detección de artefactos que puedan afectar a los datos, como píxeles defectuosos. Tras esto, se corrigen estos defectos. Después de este paso, se extraen las curvas de luz y se eliminan las estrellas que no son objeto de interés para el estudio correspondiente. En esta investigación se han tomado estos datos, con las curvas de luz ya individualizadas.

El procedimiento llevado a cabo no presenta grandes diferencias con el realizado en Koch et al. [12] y por Bryson et al en 2010 [13], salvo en este próximo paso. Una vez aisladas las curvas de luz, se ha pasado la señal por un filtro de mediana para corregir las tendencias sistemáticas, mientras que para este mismo fin los científicos responsables han aplicado sofisticados algoritmos, como la diferencia de fotometría de movimiento medio o el uso de una función de respuesta de píxeles.

El enfoque de la diferencia de fotometría de movimiento medio utiliza estrellas vecinas en el campo de visión del satélite como referencias para corregir las variaciones sistemáticas. Se calcula el promedio del brillo de las estrellas vecinas y se compara el resultado con el brillo de la estrella estudiada, asumiendo cualquier diferencia como un error sistemático.[12]

Por otro lado, en el uso de una función de respuesta de píxeles se modela cómo la luz de una estrella se distribuye en los píxeles del detector. Esto permite corregir los cambios en la respuesta del detector, como variaciones en la sensibilidad y el enfoque.[13]

Como se ha indicado con anterioridad, para filtrar la señal y eliminar el ruido se ha optado por un filtro de mediana, ya que al ser un ruido no gaussiano, este filtro no se ve tan expuesto a los *outliers* como otros filtros de paso bajo y es un proceso mucho más simple que cualquiera de los algoritmos descritos anteriormente. Esto se debe a que en lugar de guardar la media de un conjunto de datos, guarda la mediana, siendo un gran método para filtrar señales con interferencias o con ruido no gaussiano debido a su robustez frente a los *outliers*. También es cierto que debido a este mismo motivo, distorsiona los datos de los extremos. Esto se debe a que al no poder coger suficientes datos por la izquierda (o por la derecha), el valor de la mediana estará sesgado. Este filtro se ha implementado en el *listing* 8 mediante el uso de la función “medfilt” del paquete “signal” de SciPy.

Una vez eliminada la mayor parte del ruido, el siguiente paso en ambos procesos es la eliminación de fuentes de ruido conocidas; en el caso desarrollado en esta investigación sería la eliminación de la posible variabilidad de la curva de luz de la estrella. Esto podría extenderse a cualquier otro fenómeno astrofísico distinto de los tránsitos planetarios.

A continuación, se procede a la caracterización del posible exoplaneta. Se determina el periodo, el radio y otros parámetros. Finalmente, los posibles exoplanetas se someten a una serie de pruebas y análisis adicionales para confirmar su autenticidad.

### La variabilidad de la curva de luz de la estrella

Una vez filtrados los datos y eliminado el ruido, se ha estudiado la variabilidad de la curva de luz de la estrella, ajustándola a una función seno del tipo

$$V(t, C, P_v, \varphi, O) = C \sin \left( 2\pi \frac{t}{P_v} + \varphi \right) + O, \quad (2.2)$$

donde  $C$  es la amplitud del seno,  $P_v$  el periodo,  $\varphi$  la fase y  $O$  es el *offset*; el orden de magnitud del flujo total. Una vez caracterizada esta parte de la señal, se puede sustraer a la señal ya filtrada para eliminar esta parte variable, obteniendo el tránsito planetario puro.

Al haber eliminado todo elemento de la señal a excepción del tránsito, este será el periodo dominante, por lo que realizando un análisis del espectro de potencias mediante una transformada de Fourier se puede obtener la frecuencia del tránsito y, por ende, su periodo. Este es el objetivo último de limpiar la señal, determinar el periodo del tránsito planetario ( $P$ ).

## 2.3 Efectividad del método

Primeramente, el método se ha puesto a prueba con los datos simulados, de los cuales se conocen todos los parámetros de la variabilidad estelar y de los tránsitos, entre ellos, el periodo. La curva de luz resultante se ha presentado anteriormente en la figura 2.2. Tras limpiar los datos se ha obtenido la figura 2.3, pasando a determinar su periodo, coincidiendo con el periodo introducido como *input*.

Para conocer los puntos fuertes y débiles de este método se ha estudiado el funcionamiento del código desarrollado, sometiéndolo a dos pruebas: contrastar los valores obtenidos para el periodo mediante el uso del *software* Period04 con los datos originales y los datos limpios y comprobando la dependencia del error relativo frente a ciertos parámetros.

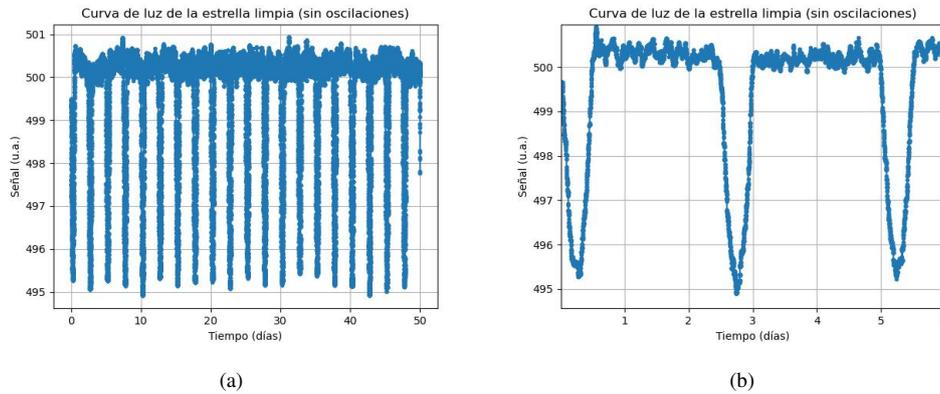


Figura 2.3: Curva de luz de la estrella simulada una vez se le ha eliminado la mayor parte del ruido, mediante el uso de un filtro de mediana, y la variabilidad. Se puede apreciar su parecido con la figura 2.1. En 2.3(b) se aprecian los tránsitos similares a los de la figura 2.1(b).

Mediante el uso del *software* [Period04](http://period04.net/)<sup>12</sup> se ha determinado la utilidad de implementar el método de limpieza de datos de la señal, obteniendo el periodo del tránsito planetario sin aplicar el método de limpieza y aplicándolo. La idea era estudiar las diferencias entre los valores del periodo que se ha obtenido limpiando y sin limpiar los datos con respecto al valor bibliográfico o el *input* (figura 2.4).

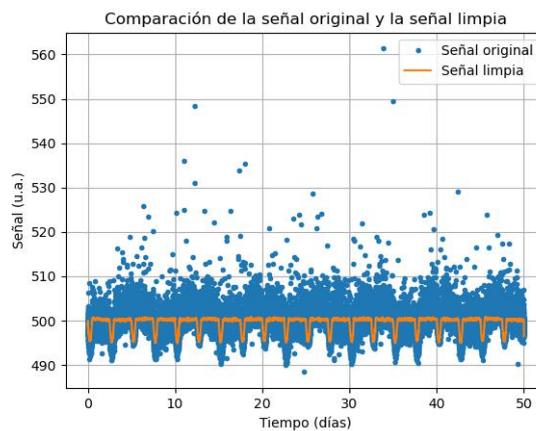


Figura 2.4: Comparación de la señal limpia y la señal original. Se aprecia que el periodo del tránsito es el mismo, al igual que una disminución de la profundidad del tránsito debido al uso del filtro de mediana. Esto se debe a que el filtro de mediana elimina los detalles más finos.

Para determinar la fiabilidad del método de limpieza de datos se ha estudiado la dependencia del error relativo del periodo respecto a la variación de ciertos parámetros: del tránsito planetario puro (sin ruido ni variabilidad estelar), del ruido y de la variabilidad de la estrella. Para cada caso se ha promediado el valor obtenido en mil simulaciones para cada punto. El código empleado puede verse en los *listings* 2, 3, 4, 13, 14, 15 y 16. Se ha trabajado bajo la idea de calcular la discrepancia porcentual (ver ecuación 2.3) entre el periodo obtenido al aplicar el

<sup>1</sup><http://period04.net/>

<sup>2</sup>El *software* Period04 es un programa que mediante la FFT encuentra repeticiones periódicas de datos a partir de su espectro de potencias. El programa da al usuario la opción de limpiar distintas frecuencias hasta que encuentre la frecuencia deseada.

método de limpieza y el periodo dado al variar parámetros como la profundidad y periodo del tránsito, la amplitud, el periodo y la fase de la variabilidad estelar o la cantidad de ruido. Para ello se ha variado cada parámetro, generando un nuevo flujo cien veces para cada parámetro y calculando el error relativo en cada iteración para después obtener el valor medio del error relativo.

$$\varepsilon = \frac{|P_c - P|}{P}, \quad (2.3)$$

donde  $\varepsilon$  es la discrepancia relativa (o error relativo),  $P$  es el periodo orbital introducido y  $P_c$  es el periodo orbital calculado.

## 2.4 Caracterización del exoplaneta

Una vez conocido el tránsito planetario, es inmediato obtener algunos de los parámetros característicos del exoplaneta estudiado, siempre que se conozcan ciertas características de la estrella como su radio, masa y/o temperatura. La distancia entre la estrella y el exoplaneta (el semieje mayor),  $a$ , en unidades astronómicas (U.A.)<sup>3</sup> viene determinado por la ecuación 2.4. Partiendo de la tercera ley de Kepler (Johannes Kepler, el astrónomo renacentista):

$$\frac{P^2}{a^3} = \frac{4\pi^2}{G \cdot M_{\text{total}}},$$

donde  $G$  es la constante gravitatoria universal ( $G = 6.67430 \cdot 10^{-11} \frac{\text{N} \cdot \text{m}^2}{\text{kg}^2}$ ) y  $M_{\text{total}}$  es la masa total del sistema ( $M_{\text{total}} = M_* + M_p$ ), pero dado que la masa de la estrella ( $M_*$ ) es mucho mayor que la masa del planeta ( $M_p$ ),  $M_* \gg M_p$ , por lo que se puede aproximar la masa total del sistema como  $M_{\text{total}} \approx M_*$ . La masa hay que introducirla en unidades de masas solares;  $M_{\odot}$ ).

$$\Rightarrow a^3 \approx \left( \frac{P}{2\pi} \right)^2 \cdot G \cdot M_*,$$

$$\Rightarrow a \approx \sqrt[3]{\left( \frac{P}{2\pi} \right)^2 \cdot G \cdot M_*}, \quad (2.4)$$

El radio del exoplaneta ( $R_p$ ) se puede estimar a partir de la ecuación 2.5, ecuación 6 en Seager y Mallen-Ornelas, 2003[14], conociendo el radio del planeta ( $R_*$ ).

Asumiendo que la señal del momento en el que no hay un tránsito (denotada con el subíndice nt),  $S_{\text{nt}}$ , es proporcional al área emisora de luz,  $A_{\text{nt}} = \pi R_*^2$ , y que la señal del tránsito (denotada con el subíndice t),  $S_t$ , es proporcional al área emisora de luz en ese momento,  $A_t = \pi(R_*^2 - R_p^2)$ , la relación entre áreas es

<sup>3</sup>1 U.A. =  $149.6 \cdot 10^6$  km.

$$\frac{A_t}{A_{nt}} = \frac{\pi(R_*^2 - R_p^2)}{\pi R_*^2} = 1 - \left(\frac{R_p}{R_*}\right)^2,$$

$$\Rightarrow \frac{S_t}{S_{nt}} = \frac{k \cdot A_t}{k \cdot A_{nt}} = \frac{A_t}{A_{nt}},$$

donde  $k$  es la constante de proporcionalidad señal-área ( $S_i = k \cdot A_i$ ).

$$\Rightarrow \frac{S_t}{S_{nt}} = 1 - \left(\frac{R_p}{R_*}\right)^2,$$

$$\Rightarrow \frac{S_t}{S_{nt}} - 1 = \frac{S_t - S_{nt}}{S_{nt}} = - \left(\frac{R_p}{R_*}\right)^2,$$

$$\Rightarrow \frac{S_{\text{no tránsito}} - S_{\text{tránsito}}}{S_{\text{no tránsito}}} = \left(\frac{R_p}{R_*}\right)^2,$$

$$\Rightarrow R_p = \sqrt{\frac{S_{nt} - S_t}{S_{nt}}} \cdot R_*, \quad (2.5)$$

Finalmente, conociendo la temperatura de la estrella,  $T_*$ , (en Kelvin; K) y la distancia a la estrella (obtenida mediante el uso de la ecuación 2.4) se puede calcular la temperatura del exoplaneta ( $T_p$ ) a través de la ecuación 2.6, que se deriva de la siguiente manera.

Se sabe que la variación de la energía recibida por un planeta respecto al tiempo viene dada por la siguiente ecuación:

$$\left(\frac{dE}{dt}\right)_{\text{recibida}} = F_{\text{rad}} \pi R_p^2,$$

donde  $R_p$  es el radio del planeta y  $F_{\text{rad}}$  viene dada por

$$F_{\text{rad}} = \frac{L}{4\pi a^2},$$

con  $a$  siendo el semieje mayor de la órbita y siendo  $L$  la luminosidad de la estrella, que se define como

$$L = 4\pi \sigma_{\text{SB}} R_*^2 T_*^4,$$

donde  $\sigma_{\text{SB}}$  es la constante de Stefan-Boltzmann.

También se sabe que la variación de la energía emitida por un planeta respecto al tiempo viene dada por esta otra ecuación:

$$\left(\frac{dE}{dt}\right)_{\text{emitida}} = 4\pi\sigma_{\text{SB}}R_p^2T_p^2.$$

Asumiendo un que el planeta emite igual que un cuerpo negro:

$$\left(\frac{dE}{dt}\right)_{\text{recibida}} = \left(\frac{dE}{dt}\right)_{\text{emitida}}.$$

Despejando la temperatura del exoplaneta se obtiene la ecuación 2.6:

$$\begin{aligned} \frac{4\pi\sigma_{\text{SB}}R_*^2T_*^4}{4\pi a^2}\pi R_p^2 &= 4\pi\sigma_{\text{SB}}R_p^2T_p^4, \\ \Rightarrow \frac{\sigma_{\text{SB}}R_*^2T_*^4}{a^2}\pi R_p^2 &= 4\pi\sigma_{\text{SB}}R_p^2T_p^4, \\ \Rightarrow \frac{R_*^2T_*^4}{a^2} &= 4T_p^4, \\ \Rightarrow \left(\frac{T_p}{T_*}\right)^4 &= \frac{R_*^2}{4a^2}, \\ \Rightarrow \frac{T_p}{T_*} &= \sqrt[4]{\frac{R_*^2}{4a^2}} = \sqrt{\frac{R_*}{2a}}, \\ \Rightarrow T_p &= T_* \cdot \sqrt{\frac{R_*}{2 \cdot a}}. \end{aligned} \tag{2.6}$$

Los valores del radio, la masa y la temperatura de cada estrella se han obtenido del catálogo [Kepler Objects of Interest](#).

# 3

## Resultados

En este capítulo se presentan los principales resultados obtenidos. El capítulo se divide en dos partes, la primera está orientada a la presentación de los resultados obtenidos mediante el uso de simulaciones, mientras que la segunda presenta los resultados al aplicar el código a objetos estudiado por el satélite Kepler.

### 3.1 Resultados de las simulaciones

Mediante el uso de los códigos desarrollados, se ha conseguido simular el tránsito de un exoplaneta y limpiar los datos de su serie temporal. Para ello, en primer lugar se ha simulado un tránsito puro (figura 2.1) mediante el uso de la ecuación 1.2. Para esta simulación, el programa pedirá al usuario algunos parámetros, como el periodo del tránsito, la duración del mismo, en que punto se encuentra el primer mínimo, la profundidad del tránsito y la duración del intervalo de observación. Para la simulación aquí presentada se han utilizado los parámetros descritos en la Tabla 3.1, donde  $t_{period}$  es el periodo del tránsito (en días),  $t_{duration}$  es la duración del mismo (también en días),  $t_0$  es la posición temporal del primer mínimo,  $depth$  es la profundidad del tránsito y, finalmente,  $t_{start}$  y  $t_{end}$  son los tiempos inicial y final de observación.

Parámetro	$t_{period}$ [días]	$t_{duration}$ [días]	$t_0$ [días]	$depth$ [u. a.]	$t_{start}$ [días]	$t_{end}$ [días]
Valor	2.5	0.5	0.25	5	0	50

Tabla 3.1: Parámetros del tránsito planetario simulado.

Tras esta primera simulación, se ha simulado una variabilidad estelar sinusoidal (mediante ecuación 2.2) y un ruido no gaussiano mediante la suma de un ruido gaussiano y un ruido debido a una distribución  $t$  no central (ecuación 2.1).[5]

Sumando los conjuntos de datos obtenidos con estas funciones a los datos del tránsito puro (ver los datos simulados en la figura 2.2). Para poder simular el

ruido no gaussiano y la variabilidad estelar, el programa solicita al usuario introducir ciertos parámetros. Para la simulación aquí presentada se han utilizado los parámetros descritos en la Tabla 3.2, donde  $A$  es la fracción de *outliers*,  $\nu$  es el número de grados de libertad,  $\delta$  es el parámetro no central de la distribución,  $\mu$  es el valor medio de la distribución y  $\sigma$  su desviación estándar. Por otro lado,  $C$  es la amplitud de la función sinusoidal que define la variabilidad estelar,  $P_\nu$  es su periodo,  $\phi$  su fase y  $O$  su *offset*. Finalmente,  $R_*$ ,  $M_*$  y  $T_*$  son el radio, la masa y la temperatura de la estrella, respectivamente. Estos parámetros son solicitados para la caracterización del exoplaneta simulado.

Parámetros ruido	Valor	Parámetros variabilidad	Valor	Parámetros estelares	Valor
$A$	0.3	$C$ [u. a.]	2	$R_*$ [ $\odot$ ]	1
$\nu$	3	$P_\nu$ [días]	7	$M_*$ [ $M_\odot$ ]	1
$\delta$	1	$\phi$ [ $\pi$ rad]	1	$T_*$ [K]	6000
$\mu$	0	$O$ [u. a.]	500		
$\sigma$	1				

Tabla 3.2: Parámetros del ruido, de la variabilidad estelar y de la estrella. Se ha escogido  $\nu = 3$  para que  $f_{\text{NCT}}$  sea perfectamente distinguible de  $f_{\text{ND}}$ .

Una vez se han simulado los datos, se les ha aplicado un filtro de mediana proveniente del paquete SciPy de Python. En la figura 3.1 se muestran los datos filtrados (Subfigura 3.1(a)) y una comparación de estos mismos con los datos originales (Subfigura 3.1(b)).

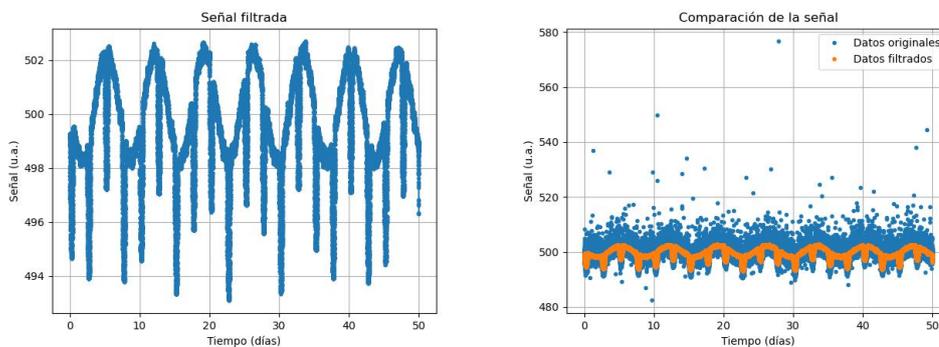


Figura 3.1: En 3.1(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.1(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino.

Una vez se ha conseguido eliminar el ruido de la señal, o al menos la mayor parte, se ha ajustado la señal filtrada a una función que defina la variabilidad de la estrella (figura 3.2), de esta manera se puede simular una señal con estos valores y sustraerla de la señal filtrada; la idea es aislar la señal del tránsito puro que se ha presentado en la figura 2.1. La diferencia de estas dos señales puede verse en la figura 2.3. El mayor inconveniente de este método es que se necesita una buena estimación inicial de la amplitud, periodo, fase y *offset* de la variabilidad estelar.

Sustrayendo la variabilidad estelar se obtiene la figura 2.3, presentada anterior-

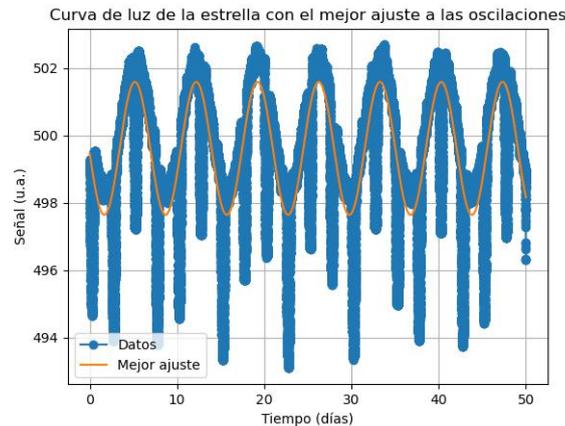


Figura 3.2: Representación del ajuste de la variabilidad sobre los datos simulados una vez pasados por el filtro de mediana.

mente. Como se puede observar, las figuras 2.1 y 2.3 se parecen, pero no son exactamente iguales. Esto no suponen ningún impedimento, pues la idea con la que se ha trabajado era la de eliminar cualquier relación periódica a excepción de los tránsitos, lo que permite realizar un periodograma<sup>1</sup> mediante el cálculo de un espectro de potencias y así poder obtener la frecuencia del máximo, o lo que es equivalente, el periodo del tránsito planetario que se busca determinar. En la figura 3.3 se puede observar el pico del espectro de potencia y todos sus múltiplos presentes en el periodograma.

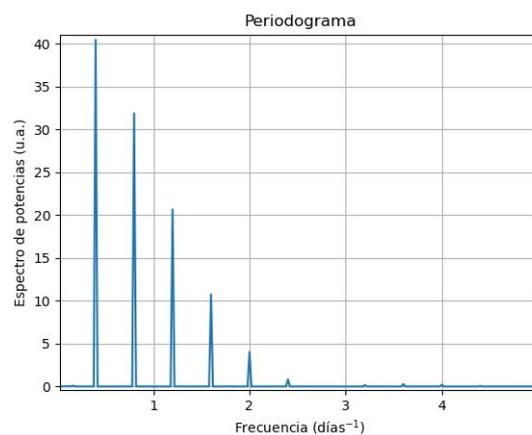


Figura 3.3: Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia  $f_{\text{dom}} = 0.4 \text{ días}^{-1}$ , lo cual se corresponde con un periodo orbital  $P = f_{\text{dom}}^{-1} = 2.5 \text{ días}$ .

Finalmente, el programa solicita al usuario la masa, el radio y la temperatura de la estrella en torno a la cual orbita el exoplaneta de la simulación. Con estos datos y el periodo se calcula el radio del planeta, la distancia entre el planeta y la estrella, la inclinación de la órbita y la temperatura del exoplaneta simulado, imprimiéndose estos resultados en pantalla.

<sup>1</sup>Un periodograma es una estimación de la densidad espectral de una señal. En términos prácticos, es la representación del espectro de potencias frente a la frecuencia con la que se repite la señal.

Los resultados se muestran en la Tabla 3.3. Dado que el único parámetro que se puede utilizar para comprobar la eficiencia del método es el periodo del planeta, se puede considerar que el método funciona perfectamente al haber obtenido el mismo periodo que el introducido a la hora de simular los datos.

	Resultado
$P$ [días]	2.50
$a$ [ $10^6$ km]	64.16
$R_p$ [ $R_{\oplus}$ ]	23.99
$i$ [ $10^{-3}\pi$ rad]	50
$T_p$ [K]	441.80

Tabla 3.3: Parámetros del exoplaneta simulado con los datos de las Tablas 3.1 y 3.2.

### Variación del error relativo

Una vez se ha demostrado que se obtiene un resultado, se ha comprobado como de efectivo es este método. Se ha estudiado la variación del error relativo en el periodo frente a la variación de distintos parámetros.

Se ha determinado que parámetros como la amplitud (figura 3.4) o la fase de la variabilidad (figura 3.5) apenas tienen influencia a la hora de determinar el periodo del tránsito.

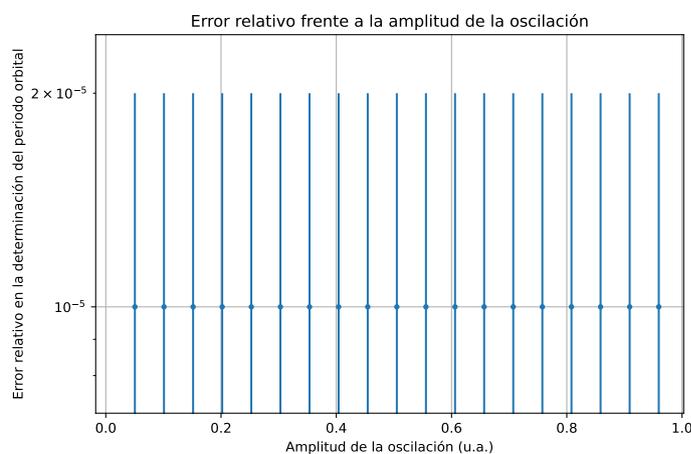


Figura 3.4: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la amplitud de la variabilidad. El eje de ordenadas (eje Y) se muestra en escala logarítmica.

En el caso de variar la profundidad del tránsito, se ha observado que el error relativo disminuye según aumenta la profundidad del tránsito, llegando a estabilizarse en torno al 0.001 % (figura 3.6), siendo este punto de estabilización una profundidad del tránsito que sea 0.4 veces la amplitud de la variabilidad, aun así, a partir de un ratio profundidad/amplitud igual a 0.3 el error relativo no supera el

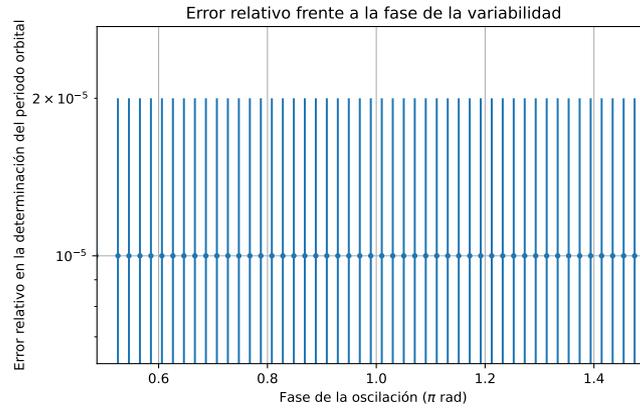


Figura 3.5: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la fase de la variabilidad. El eje de ordenadas (eje  $Y$ ) se muestra en escala logarítmica.

1.5%.

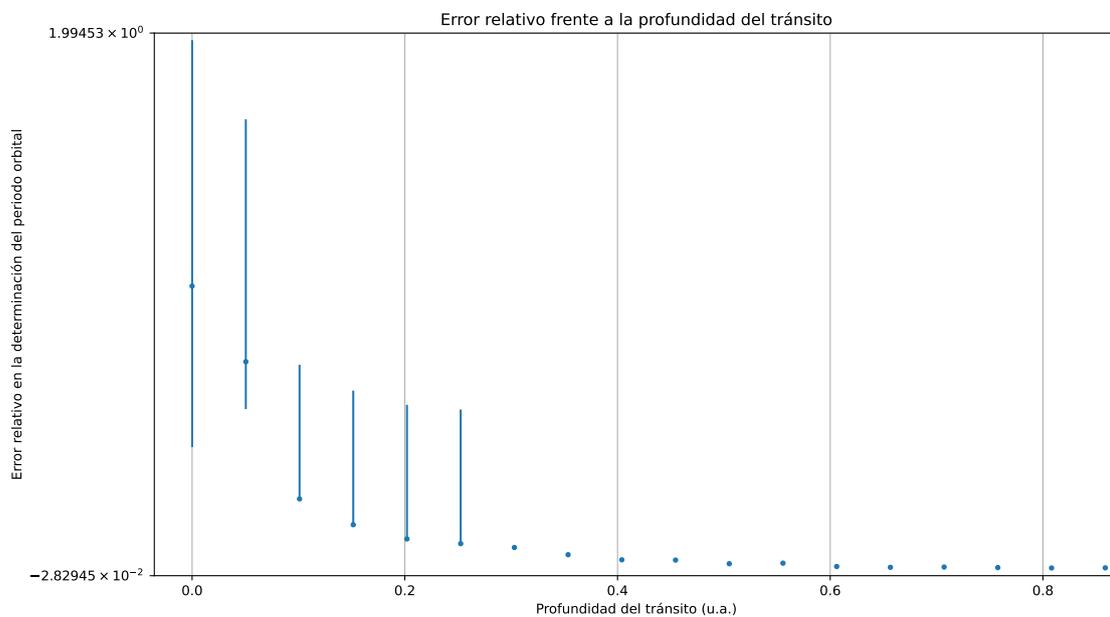


Figura 3.6: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la profundidad del tránsito. El eje de ordenadas (eje  $Y$ ) se muestra en escala logarítmica.

Por otro lado, el error relativo sufre una dependencia con el aumento del ruido de la señal. A partir de 4.8 veces el ruido generado con los parámetros de la Tabla 3.2, el error relativo comienza a aumentar, sin llegar a superar, dentro del rango parámetros que se ha considerado, el 2% de error (figura 3.8).

En el caso de la variación del periodo del tránsito (figura 3.8), el error relativo sufre un comportamiento un tanto extraño. No se corresponde con una curva creciente ni decreciente, sino más bien oscilante. Esto puede deberse a que,

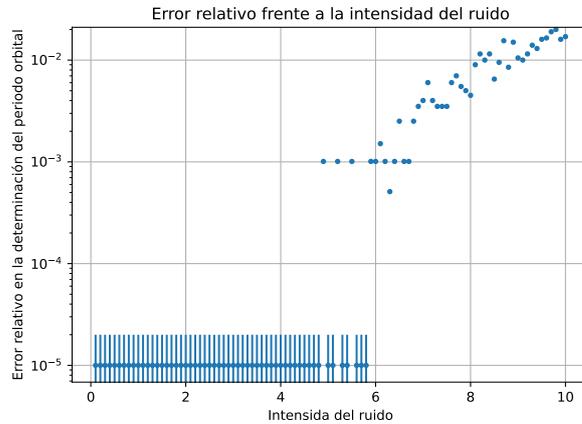


Figura 3.7: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la cantidad de ruido. El eje de ordenadas (eje  $Y$ ) se muestra en escala logarítmica.

para periodos muy bajos, el ajuste de la variabilidad no sea lo suficientemente bueno y la mantenga, por lo que la frecuencia dominante sería la de esta variabilidad sinusoidal. En el caso de periodos altos, podría ser que el programa tome como frecuencia dominante la del ruido residual, tras pasar el filtro de mediana y eliminar la variabilidad de la estrella, en lugar de los tránsitos, ya que son mucho menos frecuentes.

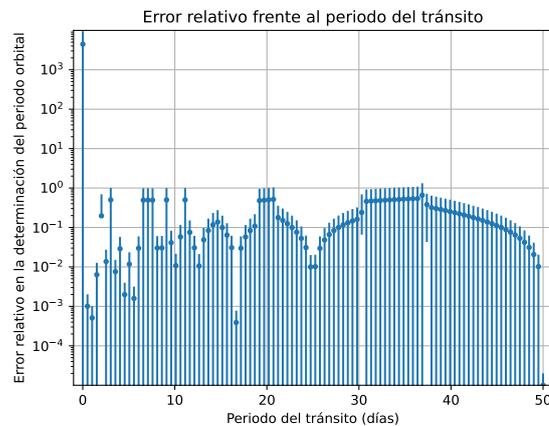


Figura 3.8: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación del periodo del tránsito. El eje de ordenadas (eje  $Y$ ) se muestra en escala logarítmica.

Finalmente, ha quedado clara la extrema dependencia del error relativo frente al periodo de la variabilidad estelar. Esta dependencia es tan fuerte que, si el periodo de la variabilidad generada no se aproxima lo suficiente al valor dado como parámetro óptimo inicial, la función “`curve_fit()`” de SciPy es incapaz de ajustar la curva sinusoidal. Para mostrar la importancia de partir de un buen parámetro óptimo inicial se ha decidido eliminar estos parámetros y dar libertad a la función para que encuentre el mejor ajuste. El resultado se muestra en la figura 3.9. La dificultad a la hora de ajustar datos a una función periódica es un hecho bien conocido en el mundo del *signal processing*, ya que estas presentan múltiples mínimos locales.

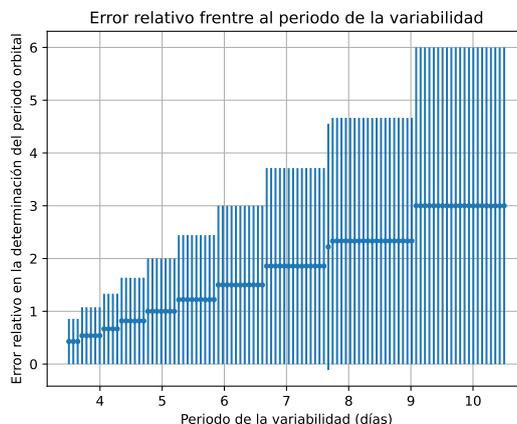


Figura 3.9: Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación del periodo de la variabilidad estelar.

## 3.2 Aplicaciones a exoplanetas reales detectados por Kepler

En una primera aproximación se ha optado por trabajar con los exoplanetas TrES-2 b y Kepler-75 b, pues están perfectamente estudiados y son candidatos ideales para testar la validez del trabajo desarrollado. Los datos se han obtenido de [Kepler Objects of Interest](#), seleccionando entre los datos de cadencia corta (*Short Cadence*) de cada exoplaneta estudiado. Esto último es de vital importancia, pues al emplear menos tiempo por imagen, hay una mayor cantidad de imágenes y, por tanto, un mayor número de puntos. Para poder obtener el error de los parámetros que caracterizan el exoplaneta es necesario descargar no solo los datos, sino el archivo con los errores correspondiente a cada paquete de datos y nombrarlo de la misma manera, añadiendo “\_err” justo antes del formato del archivo.

### TrES-2 b

En primer lugar, usando el mismo código que el utilizado para simular y analizar en la sección anterior se introduce el nombre del archivo que se quiere leer, en este caso el archivo se llama “TrES2b.tbl”, siendo los datos los representados en la figura 3.10.

Una vez se han leído los datos del exoplaneta TrES-2 b (Kepler-1 b), se les ha aplicado el mismo filtro de mediana que el usado en la sección anterior. En la figura 3.11 se muestran los datos filtrados (Subfigura 3.11(a)) y una comparación de estos mismos con los datos originales (Subfigura 3.11(b)).

En la figura 3.11(b) se observa que, debido al filtro de mediana, se pierde en torno a un 37.8% de la profundidad del tránsito. Esto se debe, como se ha mencionado anteriormente, a que el filtro de mediana elimina los detalles más finos, en este caso, reduce la profundidad del tránsito. Por este motivo, se ha optado por utilizar los datos filtrados únicamente (en los casos en los que no hay un ruido excesivo y los tránsitos están claramente marcados) para la determinación del

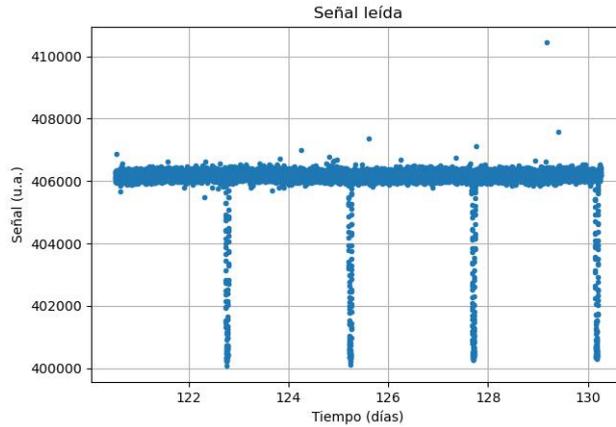


Figura 3.10: Datos correspondientes a la curva de luz de la estrella TrES-2, obtenidos del catálogo *Kepler Objects of Interest*. En este caso, los tránsitos planetarios están fuertemente marcados.

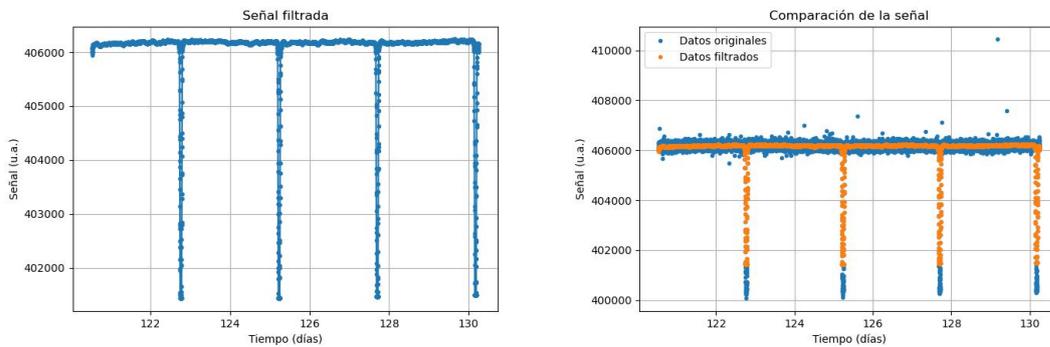


Figura 3.11: En 3.11(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.11(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino.

periodo y usar los datos originales para la determinación del resto de parámetros del exoplaneta.

Una vez se ha conseguido disminuir el ruido de la señal, se ha ajustado la señal filtrada a una función que defina la variabilidad de la estrella (figura 3.12), aunque en este caso no hay variabilidad estelar y el tránsito es bastante claro). La diferencia de estas dos señales puede verse en la figura 3.13. Esta aproximación se ha realizado con los mismos parámetros iniciales que en la sección anterior, por lo que quizá no es tan importante una buena primera aproximación.

Tras la limpieza de datos, se ha realizado un periodograma mediante el cálculo de un espectro de potencias. En la figura 3.14 se puede observar el pico del espectro de potencia, junto al resto de frecuencias y sus correspondientes alias.

Finalmente, el programa solicita al usuario la masa ( $M_*$ ), el radio ( $R_*$ ) y la temperatura ( $T_*$ ) de la estrella en torno a la cual orbita el exoplaneta estudiado. Con estos datos y el periodo se ha calculado el radio del planeta, la distancia entre



Figura 3.12: Señal filtrada con la curva de luz que mejor se ajusta a la variabilidad de la estrella. En este caso se aprecia que se trata de una oscilación con amplitud nula.

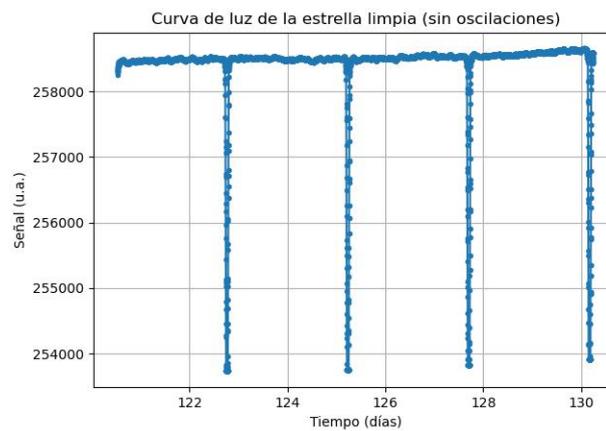


Figura 3.13: Curva de luz de la estrella una vez se le ha eliminado la mayor parte del ruido.

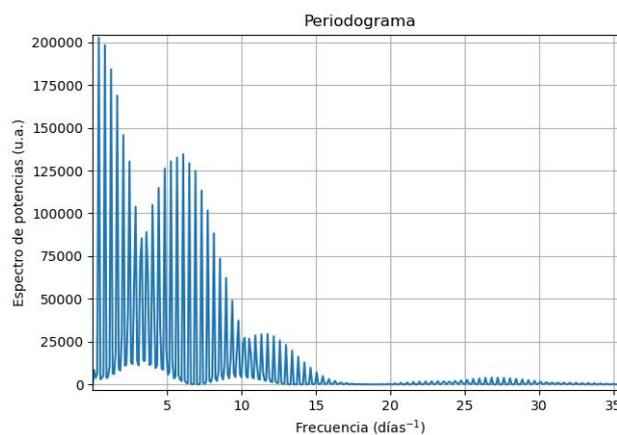


Figura 3.14: Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia  $f_{\text{dom}} \approx 0.4125 \text{ días}^{-1}$ , lo cual se corresponde con un periodo orbital  $P = f_{\text{dom}}^{-1} = 2.4241 \text{ días}$ .

el planeta y la estrella, la inclinación de la órbita y la temperatura del exoplaneta

TrES-2 b, imprimiéndose estos resultados en pantalla. Los resultados obtenidos para el exoplaneta TrES-2 b tras introducir los parámetros estelares (Tabla 3.4) se muestran en la Tabla 3.5, junto a los resultados bibliográficos obtenidos.[15]

$R_*$ [ $R_\odot$ ]	$M_*$ [ $M_\odot$ ]	T [K]
$0.964 \pm 0.038$	$1.08^{+0.11}_{-0.05}$	$5820 \pm 78$

Tabla 3.4: Parámetros estelares de la estrella TrES-2.

	Resultado	Incertidumbre	Bibliografía [15]	Discrepancia
$P$ [días]	2.4241	0.0005	$2.470613501 \pm 0.000000058$	2 %
$a$ [ $10^6$ km]	64	3	$5.32 \pm 0.11$	1103 %
$R_p$ [ $R_\oplus$ ]	12.8	0.5	$13.78 \pm 0.73$	7 %
$i$ [ $10^{-5}\pi$ rad]	273.9	0.5	$11200 \pm 200$	98 %
$T_p$ [K]	420	20	$1466 \pm 9$	71 %

Tabla 3.5: Parámetros del exoplaneta TrES-2 b para este trabajo. Los resultados se presentan junto a su incertidumbre, el valor bibliográfico y la discrepancia.

El motivo para el que la discrepancia entre el valor obtenido y el valor bibliográfico para parámetros como el semieje mayor (y, por tanto, de la temperatura) es que se ha asumido que la órbita tiene excentricidad  $e = 0$  (es una circunferencia) y que la masa de la estrella es mucho mayor que la del planeta, cuando no se tiene información de esta última. Por otro lado, el motivo para una discrepancia tan grande en el valor de la inclinación, podría deberse a que al ser números tan pequeños, cualquier mínima diferencia va a suponer una elevada discrepancia.

### Kepler-75 b

Siguiendo el mismo proceso que en el caso del exoplaneta TrES-2 b, se ha leído el fichero de datos del exoplaneta Kepler-75 b (“Kepler75b.tbl”), siendo su curva de luz la representada en la figura 3.15.

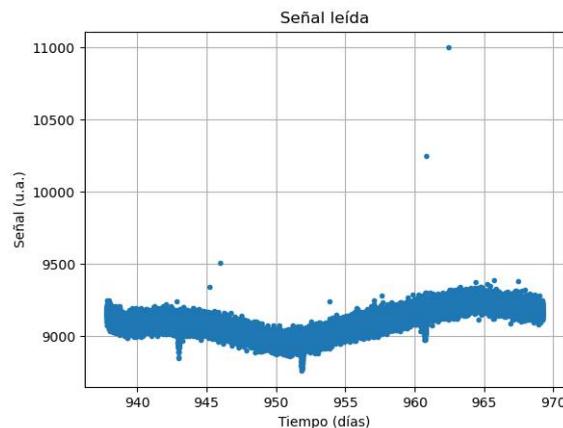


Figura 3.15: Datos correspondientes a la curva de luz de la estrella Kepler-75, obtenidos del catálogo *Kepler Objects of Interest*.

Una vez se ha obtenido la curva de luz, se ha de aplicar el filtro de mediana (figura 3.16), se realiza el mejor ajuste a la variabilidad estelar (figura 3.17) y se limpia la señal (figura 3.18). Una vez limpiados los datos, se procede a la obtención del periodo del tránsito mediante la realización de un periodograma (figura 3.19).

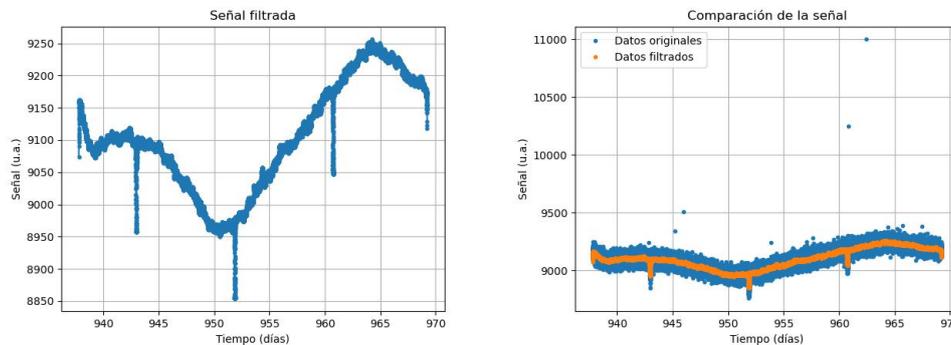


Figura 3.16: En 3.16(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.16(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino.

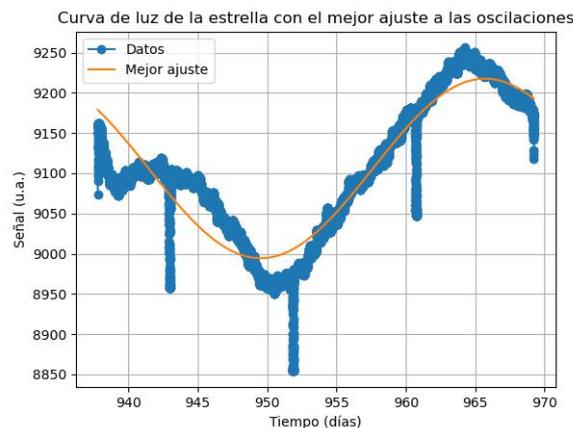


Figura 3.17: Señal filtrada con la curva de luz que mejor se ajusta a la variabilidad de la estrella. En este caso se aprecia que la variabilidad no siguen la forma sinusoidal que se esperaba. Esto ha sido una fuente de problemas.

Se puede observar que, en este caso, el mejor ajuste a la variabilidad estelar no es tan ideal como cabría de esperar. Esto se debe a que a la hora de determinar los parámetros óptimos de la curva se sigue partiendo de una semilla inicial pensada para los datos simulados con los parámetros introducidos, además de que la variabilidad estelar de la estrella en torno a la cual orbita el exoplaneta Kepler-75 b no sigue una curva sinusoidal perfecta, por lo que para las condiciones en las que se ha trabajado, es un ajuste bastante bueno, aunque no consiga eliminar la variabilidad estelar.

Los resultados obtenidos para el exoplaneta Kepler-75 b tras introducir los parámetros estelares (Tabla 3.6) se muestran en la Tabla 3.7, junto a los resultados bibliográficos obtenidos.[16]

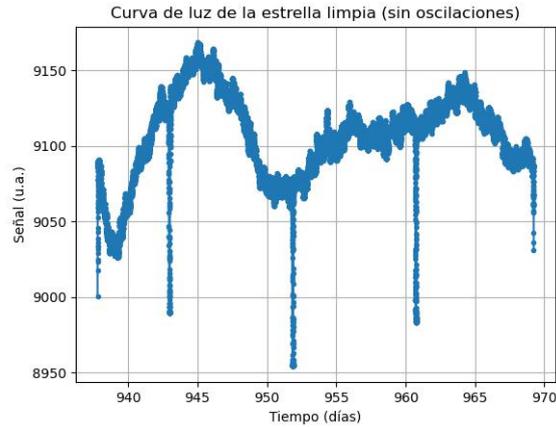


Figura 3.18: Curva de luz de la estrella una vez se le ha eliminado la mayor parte del ruido. Se ha intentado eliminar también la variabilidad de la estrella, aunque no se ha tenido mucho éxito en esto último.

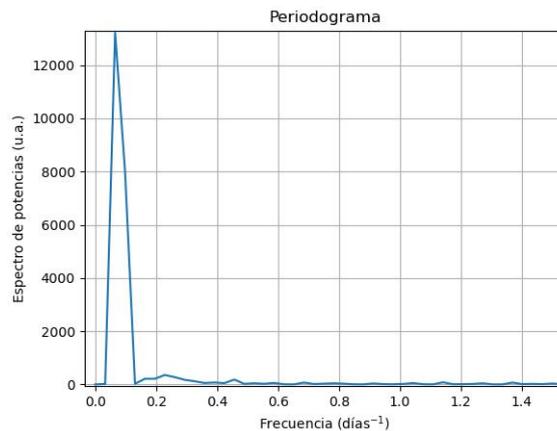


Figura 3.19: Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia  $f_{\text{dom}} \approx 0.0654 \text{ días}^{-1}$ , lo cual se corresponde con un período orbital  $P = f_{\text{dom}}^{-1} = 15.3 \text{ días}$ . Como se indica en el texto, este valor tiene una discrepancia del 77% con respecto al valor teórico.

$R_*$ [ $R_{\odot}$ ]	$M_*$ [ $M_{\odot}$ ]	$T$ [K]
$0.843 \pm 0.045$	$0.88 \pm 0.06$	$5323^{+106}_{-96}$

Tabla 3.6: Parámetros estelares de la estrella Kepler-75.

### 3.3 Period04

Viendo que en el caso del exoplaneta Kepler-75 b el resultado obtenido dista mucho del valor esperado, es sensato preguntarse si tiene alguna utilidad el código desarrollado. Para ello, se ha estudiado esta utilidad mediante el *software* [Period04](#).

Se han obtenido los periodogramas para la simulación (figura 3.20) y los exoplanetas TrES-2 b (figura 3.21) y Kepler-75 b (figura 3.22), para los datos limpios y sin limpiar.

	Resultado	Incertidumbre	Bibliografía [16]	Discrepancia
$P$ [días]	15.3	0.5	$8.9 \pm 1.6$	77%
$a$ [ $10^6$ km]	206	10	$12.24 \pm 0.18$	1616%
$R_p$ [ $R_{\oplus}$ ]	17.9	1.0	$11.985^{+0.589}_{-0.555}$	49%
$i$ [ $10^{-3} \pi$ rad]	44.0	1.5	$15^{+9}_{-11}$	300%
$T_p$ [K]	201	12	$767 \pm 15$	74%

Tabla 3.7: Parámetros del exoplaneta Kepler-75 b para este trabajo. Los resultados se presentan junto a su incertidumbre, el valor bibliográfico y la discrepancia. Se observa que el valor obtenido para el periodo dista del valor bibliográfico. no obstante, en caso de seleccionar el segundo pico del periodograma en lugar de este primero, se obtiene un periodo  $P \approx 10.84$  días, compatible con el valor bibliográfico.

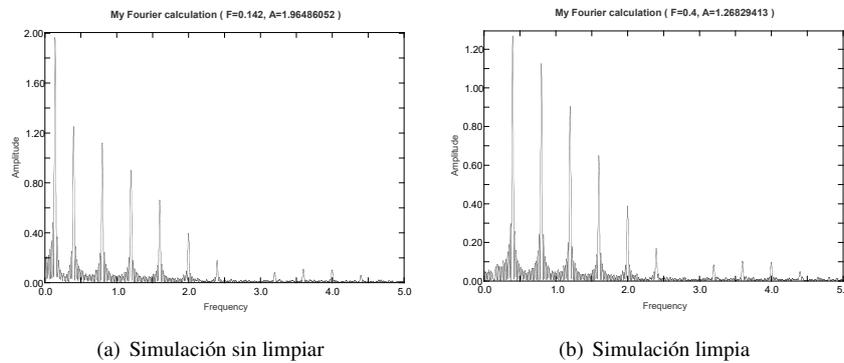


Figura 3.20: Periodogramas obtenidos, mediante el uso del *software* Period04, sin limpiar los datos, 3.20(a), y limpiando los datos, 3.20(b), para los datos simulados.

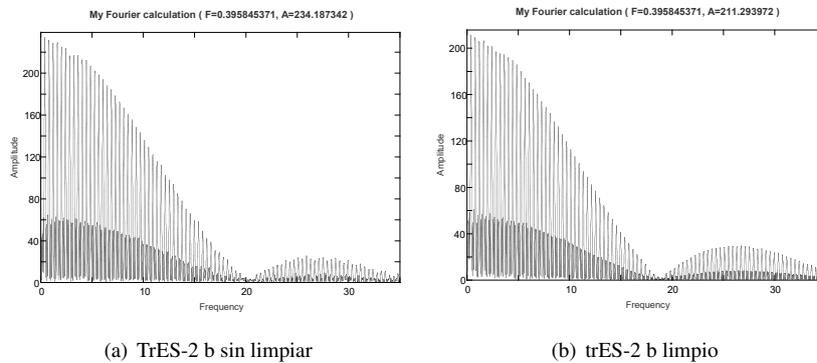


Figura 3.21: Periodogramas obtenidos, mediante el uso del *software* Period04, sin limpiar los datos, 3.21(a), y limpiando los datos, 3.21(b), para el exoplaneta TrES-2 b.

En el caso de los datos simulados, el periodo obtenido sin haber limpiado los datos es de 7.04 días, frente a los 2.50 días obtenidos al limpiar los datos, coincidiendo el resultado con el valor introducido para la simulación. En el caso de TrES-2 b, ambos periodos coinciden con un valor de 2.47 días, siendo este el valor bibliográfico en [15]. Por el contrario, para Kepler-75 b, el valor obtenido sin limpiar los datos es de 31.45 días frente a los 15.72 días obtenidos al limpiar los datos. Estos resultados no coinciden con el valor bibliográfico de [16] (8.89 días), estando el valor obtenido al limpiar los datos más cerca, con una discrepancia superior al 77%; casi el doble que el valor bibliográfico.

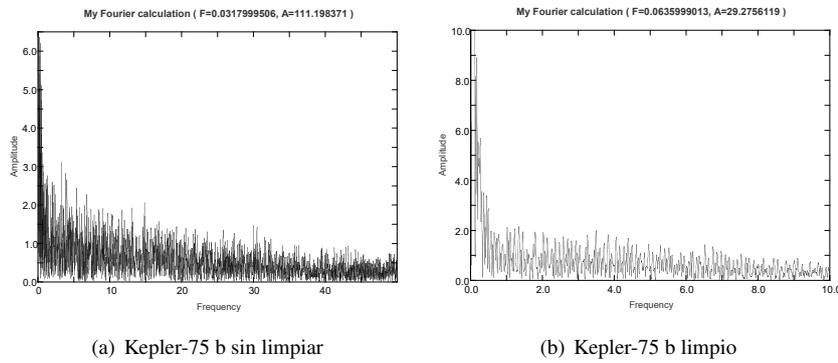


Figura 3.22: Periodogramas obtenidos, mediante el uso del *software* Period04, sin limpiar los datos, 3.22(a), y limpiando los datos, 3.22(b), para el exoplaneta Kepler-75 b.

El valor bibliográfico para este planeta se encuentra en la cuarta frecuencia detectada por el *software* Period04, como se puede ver en la Tabla 3.8. Es evidente que, a pesar de no haber conseguido eliminar todas las frecuencias subyacentes con mayor importancia que la frecuencia de los tránsitos, con el método de limpieza propuesto se ha conseguido eliminar la frecuencia dominante.

Pico	$f$ [días <sup>-1</sup> ]	$P$ [días]
1	0.0317965024	31.45
2	0.0636132315	15.72
3	0.0828350882	12.07
4	0.1126126126	8.88

Tabla 3.8: Frecuencias obtenidas por el *software* Period04 ordenadas según su dominancia en el periodograma de la figura 3.22(a), junto a los periodos correspondientes.

# 4

## Conclusiones

---

En este trabajo se han estudiado los datos provenientes del satélite Kepler, distintos tipos de ruido (no gaussiano) y varias técnicas de limpieza de datos. Se han estudiado distintas técnicas de detección y caracterización de exoplanetas. Para ello se han repasado conceptos básicos de la astronomía como la tercera ley de Kepler y se ha estudiado otros como la dependencia de la temperatura de un planeta según la temperatura a la que se encuentra su estrella y la distancia entre ambos cuerpos.

Se ha propuesto un método de limpieza de datos y caracterización de exoplanetas para las observaciones llevadas a cabo por la misión espacial Kepler. Para comprobar el funcionamiento de este método, se ha trabajado inicialmente con simulaciones de tránsitos exoplanetarios junto a su ruido (proveniente de la estrella, del propio satélite y de otros medios), quedando perfectamente caracterizado al obtener el mismo periodo que el introducido como *input*. Ante el buen resultado obtenido, se ha utilizado este método para la caracterización de los exoplanetas TrES-2 b y Kepler-75 b, encontrándose en este último un gran error relativo en la determinación del periodo de los tránsitos. El motivo de esta discrepancia no es otra que la calidad del ajuste de la variabilidad estelar a la función seno definida, haciendo que la frecuencia dominante sea la del tránsito del exoplaneta o la de la variabilidad estelar. Esto sugiere una idea de mejora futura a través del estudio de mejores métodos para tratar la variabilidad, por ejemplo mediante el uso de procesos gaussianos utilizados por Robnik y Seljak, 2020 [5][6], pues ese nivel de sofisticación está lejos del alcance de este Trabajo de Fin de Grado.

Se ha observado que los valores obtenidos para parámetros como la distancia entre los exoplanetas estudiados y las estrellas que orbitan, la inclinación de la órbita o la temperatura tienen una gran discrepancia con los valores bibliográficos. En el caso de la temperatura, estas discrepancias pueden deberse al hecho de haber tratado los exoplanetas como cuerpos negros, obviando sus atmósferas. Por otro lado, las inclinaciones no son muy pronunciadas, por lo que cualquier diferencia, a mínima que sea, representa una discrepancia importante. Finalmente, la gran discrepancia del semieje mayor ( $a$ ) puede deberse a la supo-

sición de que la órbita que describen ambos exoplanetas es una circunferencia, cuando se ha visto que no tiene por qué ser así, y este caso es solo uno entre los muchos que se pueden dar.

Para determinar el impacto que pueda tener este ajuste, se ha estudiado la dependencia del error relativo respecto a la intensidad del ruido, la amplitud, fase y periodo de la variabilidad estelar y la dependencia con la profundidad y periodo del tránsito. Se ha determinado que la influencia de parámetros como la amplitud o la fase de la variabilidad estelar apenas influyen, con un error relativo del 0.001 %. Respecto a los parámetros del ruido, la cantidad de *outliers* o la intensidad del ruido apenas tienen efecto, siendo prácticamente nulo en el caso de la cantidad de *outliers* (0.001 %) y poco importante en el caso de la intensidad del ruido, llegando hasta un máximo del 3.5 % para un ruido con desviación estándar  $\sigma = 5.8$ . Esto indica que el filtro de mediana, a pesar de ser una técnica estándar ya clásica en la literatura del procesado de señales, es una herramienta muy potente para limpiar datos con ruidos no gaussianos. La dependencia de la profundidad del tránsito es relevante mientras esta “depresión” no sea distinguible del resto de la señal, por lo tanto, a partir de un ratio profundidad/amplitud de la variabilidad superior a 0.2, el error relativo no superará el 1.5 %.

Queda entonces claro que el parámetro más determinante es el periodo de la variabilidad estelar, dependiendo prácticamente de este parámetro el ajuste sinusoidal y, por tanto, un buen ajuste. Se ha comprobado que el ajuste no sufre una gran dependencia respecto a la fase o la amplitud, siendo el periodo el parámetro más importante. Debido a este motivo, es importante tener un buen “parámetro óptimo inicial”<sup>1</sup>, facilitando el proceso de ajuste y obteniendo un mejor valor para el periodo de los tránsitos planetarios.

Dado que este es un Trabajo de Fin de Grado, hay muchas puertas que han quedado abiertas:

- La primera de ellas, sofisticar el método de limpieza de datos y caracterización de los exoplanetas, puesto que, aunque funciona aceptablemente en los casos más sencillos, en cuanto los datos se alejan del modelo propuesto, las discrepancias con los valores bibliográficos se disparan.
- Una forma de mejorar esta caracterización sería el uso de más de un paquete de datos por exoplaneta y tomando como valor de cada parámetro del exoplaneta el promedio de los valores obtenidos con cada fichero. Esto ayudaría a disminuir errores estadísticos a la hora de caracterizar los exoplanetas.
- En caso de detectar un exoplaneta en la zona de habitabilidad de su estrella, con características compatibles con la vida, podría estudiarse su composición atmosférica en busca de biomarcadores y agua. Este estudio podría realizarse determinando las líneas del espectro electromagnético absorbidas por la atmósfera, tal y como se ha hecho en Žák et al., 2019[17] y Quanz et al., 2021[18].

---

<sup>1</sup>Este “parámetro óptimo inicial” se puede modificar fácilmente en el código empleado (*Listing 10*)

- Finalmente, sería interesante también estudiar el tamaño óptimo de la ventana utilizada en el filtro de mediana (la cantidad de puntos utilizados) o la variación de los parámetros del ruido, no habiendo hecho este estudio por falta de tiempo.



## Bibliografía

---

- [1] P. Brennan, K. Walbolt, and A. Biferno, “5 ways to find a planet.”
- [2] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Señales y Sistemas*, ch. 3-6, pp. 177–514. PRENTICE HALL HISPANOAMERICANA, S.A., 1998.
- [3] R. Chassaing and D. Reay, *Digital Signal Processing and Applications with the TMS320C6713 and TM320C6416 DSK*, ch. 6, pp. 255–318. WILEY-INTERSCIENCE, 2008.
- [4] J. D. Echeverry, C. Guarnizo Lemus, and Orozco G., “Análisis de la densidad espectral de potencia en registros mer,” *Scientia et Technica*, vol. 1, ene. 2007.
- [5] J. Robnik and U. Seljak, “Kepler data analysis: Non-gaussian noise and fourier gaussian process analysis of stellar variability,” *The Astronomical Journal*, vol. 159, p. 224, apr 2020.
- [6] J. Robnik and U. Seljak, “Matched filtering with non-gaussian noise for planet transit detections,” *Monthly Notices of the Royal Astronomical Society*, vol. 504, pp. 5829–5839, apr 2021.
- [7] F. MQL5, “Distribución t.”
- [8] M. H. DeGroot and M. J. Schervish, *Probability and Statistics*, ch. 8, pp. 480–485. Addison-Wesley, 1989.
- [9] J. A. Orosz, W. F. Welsh, N. Haghighipour, B. Quarles, D. R. Short, S. M. Mills, S. Satyal, G. Torres, E. Agol, D. C. Fabrycky, D. Jontof-Hutter, G. Windmiller, T. W. A. Müller, T. C. Hinse, W. D. Cochran, M. Endl, E. B. Ford, T. Mazeh, and J. J. Lissauer, “Discovery of a third transiting planet in the kepler-47 circumbinary system,” *The Astronomical Journal*, vol. 157, p. 174, apr 2019.
- [10] M. Johnson and B. Dunbar, “Finite difference methods for diffusion processes.”
- [11] P. Chavez-Burbano, R. Sarmiento, and C. Avila, “Restauración de imágenes degradadas por ruido y distorsión lineal a través de algoritmos directos,” 07 2009.

- [12] D. G. Koch, W. J. Borucki, G. Basri, N. M. Batalha, T. M. Brown, D. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, T. N. Gautier, J. C. Geary, R. L. Gilliland, A. Gould, J. Jenkins, Y. Kondo, D. W. Latham, J. J. Lissauer, G. Marcy, D. Monet, D. Sasselov, A. Boss, D. Brownlee, J. Caldwell, A. K. Dupree, S. B. Howell, H. Kjeldsen, S. Meibom, D. Morrison, T. Owen, H. Reitsema, J. Tarter, S. T. Bryson, J. L. Dotson, P. Gazis, M. R. Haas, J. Kolodziejczak, J. F. Rowe, J. E. V. Cleve, C. Allen, H. Chandrasekaran, B. D. Clarke, J. Li, E. V. Quintana, P. Tenenbaum, J. D. Twicken, and H. Wu, "KEPLER MISSION/i DESIGN, REALIZED PHOTOMETRIC PERFORMANCE, AND EARLY SCIENCE," *The Astrophysical Journal*, vol. 713, pp. L79–L86, mar 2010.
- [13] S. Bryson, P. Tenenbaum, J. Jenkins, H. Chandrasekaran, T. Klaus, D. Caldwell, R. Gilliland, M. Haas, J. Dotson, D. Koch, and a. Borucki, "The kepler pixel response function," *The Astrophysical Journal Letters*, vol. 713, p. L97, 03 2010.
- [14] S. Seager and G. Mallen-Ornelas, "A unique solution of planet and star parameters from an extrasolar planet transit light curve," *The Astrophysical Journal*, vol. 585, pp. 1038–1055, mar 2003.
- [15] F. T. O'Donovan, D. Charbonneau, G. Mandushev, E. W. Dunham, D. W. Latham, G. Torres, A. Sozzetti, T. M. Brown, J. T. Trauger, J. A. Belmonte, M. Rabus, J. M. Almenara, R. Alonso, H. J. Deeg, G. A. Esquerdo, E. E. Falco, L. A. Hillenbrand, A. Roussanova, R. P. Stefanik, and J. N. Winn, "TrES-2: The first transiting planet in the *Kepler* field," *The Astrophysical Journal*, vol. 651, pp. L61–L64, oct 2006.
- [16] G. Hébrard, J. M. Almenara, A. Santerne, M. Deleuil, C. Damiani, A. S. Bonomo, F. Bouchy, G. Bruno, R. F. Díaz, G. Montagnier, and C. Moutou, "KOI-200 b and KOI-889 b: Two transiting exoplanets detected and characterized with Kepler, SOPHIE, and HARPS-N," *Astronomy & Astrophysics*, vol. 554, p. A114, June 2013.
- [17] J. Žák, P. Kabáth, H. M. J. Boffin, V. D. Ivanov, and M. Skarka, "High-resolution transmission spectroscopy of four hot inflated gas giant exoplanets," *The Astronomical Journal*, vol. 158, p. 120, aug 2019.
- [18] S. P. Quanz, O. Absil, W. Benz, X. Bonfils, J.-P. Berger, D. DeFRère, E. van Dishoeck, D. Ehrenreich, J. Fortney, A. Glauser, J. L. Grenfell, M. Janson, S. Kraus, O. Krause, L. Labadie, S. Lacour, M. Line, H. Linz, J. Loicq, Y. Miguel, E. Pallé, D. Queloz, H. Rauer, I. Ribas, S. Rugheimer, F. Selsis, I. Snellen, A. Sozzetti, K. R. Stapelfeldt, S. Udry, and M. Wyatt, "Atmospheric characterization of terrestrial exoplanets in the mid-infrared: biosignatures, habitability, and diversity," *Experimental Astronomy*, vol. 54, pp. 1197–1221, sep 2021.
- [19] G. Kovács, S. Zucker, and T. Mazeh, "A box-fitting algorithm in the search for periodic transits," *Astronomy & Astrophysics*, vol. 391, pp. 369–377, jul 2002.
- [20] D. Herranz and P. Vielva, "Cosmic microwave background images," *IEEE Signal Processing Magazine*, vol. 27, pp. 1–1, jan 2010.

- 
- [21] T. Barclay, D. Huber, J. F. Rowe, J. J. Fortney, C. V. Morley, E. V. Quintana, D. C. Fabrycky, G. Barentsen, S. Bloemen, J. L. Christiansen, B.-O. Demory, B. J. Fulton, J. M. Jenkins, F. Mullally, D. Ragozzine, S. E. Seader, A. Shporer, P. Tenenbaum, and S. E. Thompson, "Photometrically Derived Masses and Radii of the Planet and Star in the TrES-2 System," *The Astrophysical Journal*, vol. 761, p. 53, Dec. 2012.
- [22] L. J. Esteves, E. J. W. D. Mooij, and R. Jayawardhana, "Optical phase curves of *Kepler* exoplanets," *The Astrophysical Journal*, vol. 772, p. 51, jul 2013.
- [23] C. Rodrigo, E. Solano, and A. Bayo, "SVO Filter Profile Service Version 1.0." IVOA Working Draft 15 October 2012, Oct. 2012.
- [24] D. Herranz and C. Beltrán, *Relatividad General*, ch. 7, pp. 97–98. Universidad de Cantabria, 2022.
- [25] N. M. Batalha, J. F. Rowe, S. T. Bryson, T. Barclay, C. J. Burke, D. A. Caldwell, J. L. Christiansen, F. Mullally, S. E. Thompson, T. M. Brown, A. K. Dupree, D. C. Fabrycky, E. B. Ford, J. J. Fortney, R. L. Gilliland, H. Isaacson, D. W. Latham, G. W. Marcy, S. N. Quinn, D. Ragozzine, A. Shporer, W. J. Borucki, D. R. Ciardi, I. Gautier, Thomas N., M. R. Haas, J. M. Jenkins, D. G. Koch, J. J. Lissauer, W. Rapin, G. S. Basri, A. P. Boss, L. A. Buchhave, J. A. Carter, D. Charbonneau, J. Christensen-Dalsgaard, B. D. Clarke, W. D. Cochran, B.-O. Demory, J.-M. Desert, E. Devore, L. R. Doyle, G. A. Esquerdo, M. Everett, F. Fressin, J. C. Geary, F. R. Girouard, A. Gould, J. R. Hall, M. J. Holman, A. W. Howard, S. B. Howell, K. A. Ibrahim, K. Kinemuchi, H. Kjeldsen, T. C. Klaus, J. Li, P. W. Lucas, S. Meibom, R. L. Morris, A. Prša, E. Quintana, D. T. Sanderfer, D. Sasselov, S. E. Seader, J. C. Smith, J. H. Steffen, M. Still, M. C. Stumpe, J. C. Tarter, P. Tenenbaum, G. Torres, J. D. Twicken, K. Uddin, J. Van Cleve, L. Walkowicz, and W. F. Welsh, "Planetary Candidates Observed by *Kepler*. III. Analysis of the First 16 Months of Data," , vol. 204, p. 24, Feb. 2013.



## Código

Todos los códigos se pueden encontrar en el repositorio público de [GitHub](#)<sup>2</sup> para este Trabajo de Fin de Grado. El código no requiere demasiada memoria al trabajar con funciones, ni tampoco de un tiempo excesivo para la simulación o el análisis de los datos (apenas un minuto considerando el tiempo que se tarda en introducir los datos), aumentando considerablemente cuando se quiere estudiar la dependencia del error relativo frente a distintos parámetros (aproximadamente 25 horas por parámetro estudiado).

```
def read_data(file):
    """
    Reads time and flux data from a file.

    Parameters:
    - file (str): The name of the file containing the
                  data.

    Returns:
    - time (1D array): Time data
    - flux (1D array): Flux data.

    Plots:
    Generates a plot of the read transit flux data.
    The plot is saved in 'Plots' as 'Read Transit.pdf'
    """

    # Read the data from the file
    data = np.loadtxt(file, skiprows=3)

    # Assign the data
    time: np.array(float) = data[:, 1]
    flux: np.array(float) = data[:, 2]
```

<sup>2</sup><https://github.com/ivanvillegas7/Simulation-detection-and-characterization-of-exoplanet-transits-with-Kepler-space-mission>

```

# Plot read flux
plt.figure()
plt.plot(time, flux, marker='.', ls='none')
plt.xlabel('Time (days)')
plt.ylabel('Signal (a.u.)')
plt.title('Read Flux')
plt.grid(True)
plt.savefig('../Plots/Total Flux.pdf')

# Return the time and flux values
return(time, flux)

```

Listing 1: Función “read\_data” del *script* read.py. Esta función lee los valores de tiempo y flujo de un fichero dado para representar la serie temporal. La función devuelve estos valores en dos *arrays* unidimensionales.

```

def simulate_noise(N, params):
    """Generates noise signal according to our noise ←
        distribution = Gauss + noncentral Student's t
        N = number of realizations
        params = [A, df, nc, loc, scale] see: https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/←
        scipy.stats.nct.html
    """
    N_outliers = int(params[0]*N) #number of outliers

    Gaussian = np.random.normal(loc = 0.0, scale = 1.0, ←
        size= N - N_outliers)

    outliers = sc.stats.nct.rvs(*params[1:], size=←
        N_outliers)

    fluxi = np.concatenate([Gaussian, outliers])

    np.random.shuffle(fluxi)

    return fluxi

```

Listing 2: Función “simulate\_noise” del *script* simulate.py. Esta función simula un ruido gaussiano + t de Student. Esta función fue desarrollada por Jakkob Robnik y Uroš Seljak en [5].

```

def sinusoidal_variability(time, amplitude, period, \
    phase, offset):
    """
    Defines the sinusoidal variability of a star.

    Parameters:
    - time (np.array(float)): Array of time values.
    - amplitude (float): Amplitude of the variability.

```



```
return -f
```

Listing 4: Función “generate\_parabolic\_transit” del *script* simulate.py. Esta función lee los valores del periodo, tiempo central del tránsito, duración del tránsito, profundidad del tránsito y tiempo para simular el tránsito (parabólico) de un planeta.

```
def sim_flux(noise_params):
    """
    Simulates a transit light curve with sinusoidal
    variability and noise.

    Parameters:
    - noise_params (List[float]): list of parameters
                                  for noise generation.
      noise_params[0]: fraction of outliers in the
                      data.
      noise_params[1]: degrees of freedom for
                      non-central t distribution.
      noise_params[2]: non-central parameter for
                      non-central t distribution.
      noise_params[3]: mean value for non-central
                      t distribution.
      noise_params[4]: scale parameter for
                      non-central t distribution.

    Returns:
    - time (1D array): array of time values.
    - flux (1D array): array of simulated flux values.

    Plots:
    Generates a plot of the simulated transit data.
    The plot is saved in 'Plots' as
    'Simulated Transit.pdf'
    Generates a plot of the simulated flux data.
    The plot is saved in 'Plots' as
    'Simulated Flux.pdf'

    Uses:
    - generate_parabolic_transit()
    - sinusoidal_variability()
    - simulate_noise()
    """

    t_period: float = float(input('Transit_period:_'))
    t_duration: float = float(input('Transit_duration:_'))
    t0: float = float(input('Transit_half_time:_'))
    depth: float = float(input('Transit_depth:_'))
    t_start: float = float(input('Initial_time:_'))
```

```

t_end: float = float(input('Ending time: '))

time: np.array(float) = np.linspace(t_start, t_end, \
                                   100000)

# Generate transit model
flux_p: np.array(float)
flux_p = generate_parabolic_transit(t_period, t0, \
                                   t_duration, \
                                   depth, time)

# Plot transit model
plt.figure()
plt.plot(time, flux_p, marker='.')
plt.xlabel('Time (days)')
plt.ylabel('Signal (a.u.)')
plt.title('Simulated Transit')
plt.grid(True)
plt.savefig('../Plots/Simulated_Transit.pdf')

# Generate sinusoidal variability
amplitude: float = float(input('Sinusoidal \
variability amplitude: '))
period: float = float(input('Sinusoidal \
variability period: '))
phase: float = float(input(r'Sinusoidal \
variability phase (in units of $\pi$): '))
offset: float = float(input('Stellar flux offset (\
order of magnitude): '))

variability: np.array(float)
variability = sinusoidal_variability(time, \
                                     amplitude, \
                                     period, \
                                     phase*np.pi, \
                                     offset)

# Simulate noise
noise: np.array(float)
noise = simulate_noise(len(flux_p), noise_params)

# Combine flux with variability and noise
flux: np.array(float)
flux = flux_p + variability + noise

# Plot simulated flux
plt.figure()
plt.plot(time, flux, marker='.', ls='none')
plt.xlabel('Time (days)')

```

```

plt.ylabel('Signal (a.u.)')
plt.title('Simulated Flux')
plt.grid(True)
plt.savefig('../Plots/Total_Flux.pdf')

return(time, flux)

```

Listing 5: Función “sim\_flux” del *script* simulate.py. Esta función lee los parámetros del ruido (fracción de outliers en los datos, grados de libertad para la distribución t de Student, el parámetro no central de esta distribución, el valor medio para esta distribución y el parámetro de escala de la misma). La función pregunta el resto de parámetros y llama a las funciones definidas con anterioridad (Códigos 2, 3 y 4) para generar los datos, dibujando una gráfica del tránsito puro (ver Figura ??) y de los datos finales (ver Figura ??).

```

def data_maker():
    """
    Generates simulated time series data with a
    non-central t-distribution and an added
    outlier fraction.

    Arguments:

    - A (float): outlier fraction.
    - df (float): degrees of freedom for non-central
      t distribution.
    - nc (float): non-central parameter.
    - loc (float): mean value of non-central t
      distribution.
    - scale (float): scale of non-central t
      distribution.

    Returns:

    - time (1D array): array of time values.
    - flux (1D array): array of simulated flux values.

    Uses:

    - sim_flux()
    """

    A: float
    A = float(input('Outlier fraction: '))

    df: float
    df = float(input("Non-central t distribution ←
      degrees of freedom: "))

    nc: float

```

```

nc = float(input("Non-central parameter: "))

loc: float
loc = float(input("Non-central t distribution mean ←
value: "))

scale: float
scale = float(input("Non-central t distribution ←
scale: "))

params: List[float]
params = [A, df, nc, loc, scale]

time: np.array(float)
flux: np.array(float)
time, flux = sim_flux(params)

return(time, flux)

```

Listing 6: Función “data\_maker” del *script* simulate.py. Esta función pregunta los parámetros del ruido (fracción de *outliers* en los datos, grados de libertad para la distribución t de Student, el parámetro no central de esta distribución, el valor medio para esta distribución y el parámetro de escala de la misma) y se los pasa a la función “sim\_flux” (ver Código 5).

```

def discard_variability(time, flux, params):
    """
    Subtracts the variability of the light curve from
    the flux values. It also prints the transit
    parameters.

    Parameters:
    - time (1D array): Time values.
    - flux (1D array): Flux values.
    - params (1D array): Variability and transit
      parameters.

    Returns:
    - flux_nv (1D array): Flux values with the
      variability subtracted.

    Plots:
    Generates a plot of the filtered flux data w/o
    the stellar variability.
    The plot is saved in 'Plots' as
    'Discarded Variability.pdf'
    """

    # Variability amplitude
    A: float = params[0]

```

```

# Variability period
P_v: float = params[1]

# Variability phase
Phase_v: float = params[2]

# Compute the variability
variability: np.array(float)
variability = A*np.sin(2*np.pi*time/P_v\
                    +Phase_v*np.pi)

# Compute the flux values with the variability
# subtracted
flux_nv: np.array(float)
flux_nv = flux-variability

# Plot the star's cleaned light curve (without
# variability)
plt.figure()
plt.plot(time, flux_nv, marker='.')
plt.xlabel('Time (days)')
plt.ylabel('Signal (a.u.)')
plt.title("Star's cleaned light curve (w/o stellar ←
variability)")
plt.grid(True)
plt.savefig('../Plots/Cleaned_Light_Curve.pdf')

# Return the flux values with the variability
# subtracted
return flux_nv

```

Listing 7: Función “discard\_variability” del *script* remove.py. Esta función resta la oscilación debida a la variabilidad de la estrella obteniendo este cantidad apartir de los datos obtenidos con el uso de la función “best\_fit” del *script* analyze.py (ver Código 10) que se parasan como parámetros. La función representa gráficamente los datos limpiados.

```

def median_filter(time: np.array(float),\
                 flux: np.array(float),\
                 window_size: int):
    """
    Applies a median filter of a given window size to a
    flux array.

    Parameters:
    - time (1D array): Array of time values.
    - flux (1D array): Array of flux values.
    - window_size (int): Size of the window used for
                        the median filter.
    """

```

```

Returns:
- filtered_flux (np.array(float)): Array of
                                   filtered
                                   flux values.

Plots:
Generates a plot of the original flux data and the
filtered flux data.
The plot is saved in 'Plots' as
'Flux Comparison.pdf'
Generates a plot of the filtered flux data.
The plot is saved in 'Plots' as
'Filtered Flux.pdf'
"""

# Apply median filter to flux array
filtered_flux: np.array(float)
filtered_flux = sc.signal.medfilt(flux,\
                                   kernel_size=\
                                   window_size)

# Plot the filtered and unfiltered flux
plt.figure()
plt.plot(time, flux, marker='.',\
          label='Original_data', ls='none')
plt.plot(time, filtered_flux, marker='.',\
          label='Filtered_data', ls='none')
plt.xlabel('Time_(days)')
plt.ylabel('Signal_(a.u.)')
plt.title('Flux_comparison')
plt.legend()
plt.grid(True)
plt.savefig('../Plots/Flux_Comparison.pdf')

# Plot the filtered flux data
plt.figure()
plt.plot(time, filtered_flux, marker='.')
plt.xlabel('Time_(days)')
plt.ylabel('Signal_(a.u.)')
plt.title('Filtered_flux')
plt.grid(True)
plt.savefig('../Plots/Filtered_Flux.pdf')

return filtered_flux

```

Listing 8: Función “median\_filter” del *script* `analyze.py`. Esta función recibe los los datos de tiempo y flujo y realiza un filtro de mediana con un kernel del tamaño que se indique como parámetro. La función grafica los datos sin filtrar y filtrados para realizar una comparación visual y los datos filtrados por separado, devolviendo estos últimos.



```

1, 0]))

# Plot data and its best fit
plt.figure()
plt.plot(x, y, marker='o', label='Data')
plt.plot(x, best_fit_func(x, *popt),\
         label='Best_fit')
plt.xlabel('Time (days)')
plt.ylabel('Signal (a.u.)')
plt.title("Star's light curve w/ best fit for \
stellar variability")
plt.legend()
plt.grid(True)
plt.savefig('../Plots/Best_Fit.pdf')

return popt

```

Listing 10: Función “best\_fit” del *script* `analyze.py`. Esta función busca los aprámetros óptimos en relacion a los datos proporcionados de la función “best\_fit\_func” (ver *Listing 9*), devolviendo estos parámetros. Par a un buen ajuste, es necesario proporcionar unos buenos primeros parámetros estimados.

```

def get_period(flux, sampling_f):
    """
    Calculates the period of a periodic signal in the
    input flux data. It first calculates the
    periodogram using the SciPy signal processing
    library's periodogram function. Then, it finds the
    frequency with the highest power (i.e. the dominant
    frequency) and calculates the period corresponding
    to that frequency. Finally, it plots the
    periodogram and saves the plot as 'Periodogram.pdf'
    in the 'Plots' folder.

    Parameters:
    - flux (1D array): Flux values.
    - sampling_f (float): Sampling frequency.

    Returns:
    - period (float): The period of the periodic signal
      in the flux data.

    Plots:
    Generates a plot of the power spectrum. The plot is
    saved in 'Plots' as 'Periodogram.pdf'.
    """

    # Calculate the periodogram
    frequencies, power_spectrum =\
    sc.signal.periodogram(flux, sampling_f)

```

```

# Plot the periodogram
plt.figure()
plt.plot(frequencies, power_spectrum)
plt.grid(True)
plt.xlabel(r'Frequency (days-1)')
plt.ylabel('Power Spectrum (a.u.)')
plt.title('Periodogram')
plt.savefig('../Plots/Periodogram.pdf')

# Find the frequency with the highest power (i.e. |
# the dominant frequency)
dominant_frequency = \
frequencies[np.argmax(power_spectrum)]

# Calculate the period corresponding to the
# dominant frequency
period = 1/dominant_frequency

return period

```

Listing 11: Función “get\_period” del *script* analyze.py. Esta función calcula el periodo a partir de un espectro de potencias (graficando el espectrograma), devolviendod el periodo calculado.

```

def characterization(time, flux, period):
    """
    This function characterizes an exoplanet based on
    its light curve, taking into account the stellar
    variability.

    Parameters:
    - time (1D array): Time array of the light curve.
    - flux (1D array): Flux array of the light curve.
    - period (float): Period of exoplanet's transit.'

    Returns:
    - None
    """

    # Computes the planet's radius using transits's law
    stellar_radius: float
    stellar_radius = float(input(r"Star's radius ($R_{\odot}$): "))
    transit_depth: float = \
    (np.mean(flux) - np.min(flux))/np.mean(flux)
    planet_radius: float = \
    np.sqrt(transit_depth)*stellar_radius*6.957e5/6371

    # Computes the planet-star distance using Kepler's

```

```

# law
G = 6.67430e-11 # Gravitational constant
M_star: float
M_star = float(input(r"Star's mass ($M_\odot): "))
a = ((period*24*60*60/(2*np.pi))**2*G*M_star)\
    *(1/3)
distance_to_star = a*149.6e6 # Distance in km

# Orbital inclination determination
inclination: float
inclination = np.arccos(np.sqrt(abs(np.mean(flux)+\
    np.min(flux))/\
    (2*np.mean(flux))))/np.pi

if str(inclination)=='nan':

    inclination = 0.5

# Computes the planet's temperature
stellar_temp: float
stellar_temp =\
float(input("Star's temperature (K): "))
planet_temp = stellar_temp*np.sqrt(stellar_radius*\
    6.957e5/\
    (2*\
    distance_to_star))

if star=='none':

    # Print results
    print(f'\nPeriod: {period:.2f} days.')
    print(f"Planet's radius: {planet_radius:.2f} R_\oplus.")
    print(f'Planet-Star distance: {distance_to_star:.2f} km.')
    print(f'Orbit inclination: {inclination:.2f}\pi rad.')
    print(f"Planet's temperature: {planet_temp:.2f} K.")

else:

    err: np.array(float) =\
    np.loadtxt(f'../Data/{star}_err.tbl', skiprows=3)

    rel: np.array(float) =\
    np.mean(err)/np.mean(flux)

```

```

rel_R: float = \
float(input('Stellar radius error: '))/\
stellar_radius

rel_M: float = \
float(input('Stellar mass error: '))/M_star

rel_T: float = \
float(input('Stellar temperature error: '))/\
stellar_temp

delta_P: float = period*rel

delta_R: float = planet_radius*\
np.sqrt(rel**2/2+rel_R**2)

delta_a: float = distance_to_star*\
np.sqrt((2*rel**2+rel_M**2)/3)

delta_i: float = inclination*rel

delta_T: float = planet_temp*\
np.sqrt(rel_T**2+rel_R**2/2+\
(2*rel**2+rel_M**2)/3)

# Print results
print(f'\nPeriod: ({period:.2f})\pm{delta_P:.2f}←
      )\days.')
print(f"Planet's radius: ({planet_radius:.2f})\pm←
      {delta_R:.2f})\R\oplus.")
print(f'Planet-Star distance: ({←
      distance_to_star:.2f})\pm{delta_a:.2f})\km.')
print(f'Orbit inclination: ({inclination:.2f})\pm←
      {delta_i:.2f})\pi\rad.')
print(f"Planet's temperature: ({planet_temp:.2f}←
      )\pm{delta_T:.2f})\K.")

```

Listing 12: Función “characterization” del *script* `analyze.py`. Esta función caracteriza el planeta observado una vez se ha obtenido el periodo, conociendo los valores del radio y la masa estelares. La función imprime en pantalla el periodo del planeta, su radio, la distancia a la estrella y la inclinación de su órbita.

```

def eff_flux(depth: float, amplitude: float, \
            period: float, phase: float, \
            t_per: float, mult: float):
    """
    This function generates a flux signal by combining
    a simulated transit model, sinusoidal variability,
    and simulated noise to check how effective the used
    method is.

```

```

Parameters:
- depth (float): Depth of the transit.
- amplitude (float): Amplitude of the sinusoidal
                    variability.
- period (float): Period of the sinusoidal
                 variability.
- phase (float): Phase of the sinusoidal
               variability.
- t_per (float): Duration of the transit.
- mult (float): Multiplicative factor for the
               simulated noise.

Returns:
- time (1D array): Time array.
- flux (1D array): Effective flux signal generated
                  by combining the transit model,
                  sinusoidal variability and
                  noise.
"""
t_duration: float = t_per/5
t0: float = t_duration/2
t_start: float = 0
t_end: float = 50

time: np.array(float) = np.linspace(t_start,\
                                   t_end,\
                                   100000)

# Generate transit model
flux_p: np.array(float)
flux_p =\
simulate.generate_parabolic_transit(t_per, t0,\
                                   t_duration,\
                                   depth, time)

# Simulate noise
A: float
A = 0.3

df: float
df = 3

nc: float
nc = 1

loc: float
loc = 0

```

```

scale: float
scale = 1

params: List[float]

params = [A, df, nc, loc, scale]

noise: np.array(float)
noise = simulate.simulate_noise(len(flux_p),\
                                params)

# Generate sinusoidal variability
offset: float = 500

variability: np.array(float)
variability =\
simulate.sinusoidal_variability(time, amplitude,\
                                period,\
                                phase*np.pi,\
                                offset)

# Combine flux with variability and noise
flux: np.array(float)
flux = flux_p + variability + mult*noise

return(time, flux)

```

Listing 13: Función “eff\_flux” del *script* effectivity.py. Esta función simula los datos para comprobar la eficacia del método empleado.

```

def confidence_limits(array, cl):

    """
    For a given array, gets the lower and upper values ←
    corresponding to
    a confidence limit cl. This confidence limit ←
    takes value 0.0<=cl<=1.0
    """

    x = array.copy()
    x.sort()
    p = (1.0-cl)/2
    lower = int(p*len(x))
    upper = len(x)-lower
    return np.array([x[lower], x[upper]])

```

Listing 14: Función “confidence\_limlimits” del *script* conf\_level.py desarrollado por Diego Herranz Muñoz. Esta función obtiene el intervalo de los datos en el que se encuentra el 68% de ellos mismos.

```

def effectivity(i: int):

    """
    This function analyzes the effect of different
    parameters on the relative error in determining
    the transit period. It generates synthetic data for
    various parameter combinations, applies data
    analysis techniques and calculates the relative
    error between the known transit period and the
    determined period. The function then plots the
    relative error against the chosen parameter.

    Parameters:
    - i (int): Index to select the parameter
              combination to analyze.

    Returns:
    - None

    The function generates plots of the relative error
    against the chosen parameter and saves them as PDF
    files. The specific plots generated depend on the
    value of i:

    - i=0: Relative error vs. transit depth.
    - i=1: Relative error vs. variability amplitude.
    - i=2: Relative error vs. variability period.
    - i=3: Relative error vs. variability phase.
    - i=4: Relative error vs. transit period.
    - i=5: Relative error vs. noise intensity.

    The plots are saved in the "Plots/Effectivity"
    directory.
    """

    n: int = 100

    if i==0:

        depth: np.array(float) = np.linspace(0, 5,\
                                             n)

        amplitude: np.array(float) = 2*np.ones(n)

        period: float = 7

        phase: float = 1

        t_per: np.array(float) = 2.5*np.ones(n)

```

```
    noise: float = 1

elif i==1:

    depth: np.array(float) = 5*np.ones(n)

    amplitude: np.array(float) = np.linspace(0, 5, ←
        n)

    period: float = 7

    phase: float = 1

    t_per: np.array(float) = 2.5*np.ones(n)

    noise: float = 1

elif i==2:

    depth: np.array(float) = 5*np.ones(n)

    amplitude: np.array(float) = 2*np.ones(n)

    period: np.array(float) = np.linspace(3.5,\
                                           10.5,\
                                           n)

    phase: float = 1

    t_per: np.array(float) = 2.5*np.ones(n)

    noise: float = 1

elif i==3:

    depth: np.array(float) = 5*np.ones(n)

    amplitude: np.array(float) = 2*np.ones(n)

    period: float = 7

    phase: np.array(float) = np.linspace(0, 2, n)

    t_per: np.array(float) = 2.5*np.ones(n)

    noise: float = 1

elif i==4:
```

```
depth: np.array(float) = 5*np.ones(n)

amplitude: np.array(float) = 2*np.ones(n)

period: float = 7

phase: float = 1

t_per: np.array(float) = np.linspace(0.01, 50,\
                                     n)

noise: float = 1

elif i==5:

    depth: np.array(float) = 5*np.ones(n)

    amplitude: np.array(float) = 2*np.ones(n)

    period: float = 7

    phase: float = 1

    t_per: np.array(float) = 2.5*np.ones(n)

    noise: np.array(float) = np.linspace(0.1,10, n)

err_mean: List[float] = []

err_bar_low: List[np.array(float)] = []

err_bar_up: List[np.array(float)] = []

for j in range(n):

    err: List[float] = []

    for _ in range(1000):

        time: np.array(float)
        flux: np.array(float)

        if i==0:

            time, flux = eff_flux(depth[j],\
                                 amplitude[j],\
                                 period, phase,\
                                 t_per[j], noise)
```

```
elif i==1:

    time, flux = eff_flux(depth[j],\
                           amplitude[j],\
                           period, phase,\
                           t_per[j], noise)

elif i==2:

    time, flux = eff_flux(depth[j],\
                           amplitude[j],\
                           period[j],\
                           phase, t_per[j],\
                           noise)

elif i==3:

    time, flux = eff_flux(depth[j],\
                           amplitude[j],\
                           period,\
                           phase[j],\
                           t_per[j], noise)

elif i==4:

    time, flux = eff_flux(depth[j],\
                           amplitude[j],\
                           period, phase,\
                           t_per[j], noise)

elif i==5:

    time, flux = eff_flux(depth[j],\
                           amplitude[j],\
                           period, phase,\
                           t_per[j],\
                           noise[j])

filtered_flux: np.array(float)
filtered_flux =\
    analize.median_filter(time, flux, 101)

best_fit_params: np.array(float)
best_fit_params =\
    analize.best_fit(time, filtered_flux)

flux_nv: np.array(float)
flux_nv =\
```

```

        remove.discard_variability(time,\
                                    filtered_flux,\
                                    best_fit_params)

    result: float =\
        analyze.get_period(flux_nv,\
                            1/(time[1]-time[0]))

    err.append(np.abs(t_per[j]-result)/\
                t_per[j])

err_mean.append(np.mean(err))

err_bar =\
    conf_level.confidence_limits(err, 0.68)

err_bar_low.append(err_bar[0])

err_bar_up.append(err_bar[1])

N: np.array(np.array(float) =\
            np.array(np.array(err_bar_low),\
                      np.array(err_bar_up))

plt.figure()
plt.ylabel('Relative error on the transit period ←\
           determination')
plt.grid(True)

if i==0:

    plt.errorbar(depth, err_mean, yerr=N, fmt='.')
    plt.xlabel("Transit's depth (a.u.)")
    plt.title("Relative error against transit's ←\
              depth")
    plt.savefig('../Plots/Effectivity/REvsTD.pdf')

elif i==1:

    plt.errorbar(amplitude, err_mean, yerr=N,\
                 fmt='.')
    plt.xlabel('Variability amplitude (a.u.)')
    plt.title('Relative error against variability ←\
              amplitude')
    plt.savefig('../Plots/Effectivity/REvsVA.pdf')

elif i==2:

    plt.errorbar(period, err_mean, yerr=N, fmt='.')

```

```

plt.xlabel('Variability␣period␣(days)')
plt.title('Relative␣error␣against␣variability␣↔
          period')
plt.savefig('../Plots/Effectivity/REvsVP.pdf')

elif i==3:

plt.errorbar(phase, err_mean, yerr=N, fmt='.')
plt.xlabel(r'Variability␣phase␣( $\pi$ ␣rad)')
plt.title('Relative␣error␣against␣variability␣↔
          phase')
plt.savefig('../Plots/Effectivity/REvsVPh.pdf')

elif i==4:

plt.errorbar(t_per, err_mean, yerr=N, fmt='.')
plt.xlabel('Transit␣period␣(days)')
plt.title('Relative␣error␣against␣transit␣↔
          period')
plt.savefig('../Plots/Effectivity/REvsTP.pdf')

elif i==5:

plt.errorbar(noise, err_mean, yerr=N, fmt='.')
plt.xlabel('Noise␣intensity')
plt.title('Relative␣error␣against␣noise␣↔
          intensity')
plt.savefig('../Plots/Effectivity/REvsNI.pdf')

```

Listing 15: Función “effectivity” del *script* effectivity.py. Esta función comprueba la eficacia del método empleado.

```

def get_effectivity():
    """
    The function calls the effectivity(i) function for
    different values of i to analyze the effect of
    different parameters on the relative error in
    determining the transit period. It loops through
    the values 0 to 5 and calls the effectivity()
    function for each value.

    Parameters:
    - None

    Returns:
    - None
    """
    for i in range(6):

```

---

```
effectivity(i)
```

Listing 16: Función “get\_effectivity” del *script* effectivity.py. Esta función repite las simulaciones realizadas en el Listing 15 para cada parámetro.

```
def main():
    """
    Executes the main program for filtering and
    analyzing star's light curve.

    - It offers the possibility of checking how
      effective the proposed method is.

    - It calls the 'data_maker' function to
      generate simulated data or the
      'read_data' function to read real data
      depending if the user want to analyze
      simulated or real data. It prompts the
      user to enter a window size for the
      median filter (using the 'median_filter'
      function), and then applies the filter to the
      simulated data. It finds the best parameters
      fitting the plotted curve and computes the
      transit without the stellar variability
      (printing the values of the transit main
      parameters).

    Parameters:
    - None

    Returns:
    - None

    Uses:
    - data_maker() from simulate.py
    - read_data() from read.py
    - median_filter() from analyze.py
    - best_fit() from analyze.py
    - discard_variability() from remove.py
    - get_effectivity() from effectivity.py
    """

    option: str
    option = input('Do you want to check how effective ←
                  the method is? (Y or N): ')

    # Make sure the user chooses a valid
    # option
    while option != 'Y' and option != 'N':
```

```

print('')
print('Thats not an option. Please enter a valid option (Y or N).')
option = input("Do you want to check how effective the method is? (Y or N): ")

if option == 'Y':

    print('This will take around 18 hours.')

    option: str
    option = input('Do you want to continue? (Y or N): ')

    if option == 'Y':

        effectivity.get_effectivity()

# Define the time and flux arrays for data_maker()
time: np.array(float)
flux: np.array(float)

# Ask to know if the program has to simulate or read the data
option: str
option = input("Do you want to simulate (S) or to read (R) real data? ")

# Make sure the user chooses a valid option
while option != 'S' and option != 'R':

    print('')
    print('Thats not an option. Please enter a valid option (S or R).')
    option = input("Do you want to simulate (S) or to use real (R) data? ")

if option == 'S':

    # Generate the time and flux data for the star's light curve
    time, flux = simulate.data_maker()

    star: str = 'none'

else:

    # Ask for the file name

```

```

file: str
file = input("Which file do you want to read? ")

# Read the time and flux data for the
# star's light curve
time, flux = read.read_data(f'../Data/{file}')

star: str = file[0:len(file)-4]

# Apply a median filter to the flux data
filtered_flux: np.array(float)
window_size: int = int(input('Window size for the median filter (~101): '))
filtered_flux = analyze.median_filter(time, flux, \
                                     window_size)

# Fit the filtered flux data to a curve
best_fit_params: np.array(float)
best_fit_params = analyze.best_fit(time, \
                                   filtered_flux)

# Discard the variability from the star's light curve
flux_nv: np.array(float)
flux_nv = \
remove.discard_variability(time, filtered_flux, \
                           best_fit_params)

period: float = \
analyze.get_period(flux_nv, 1/(time[1]-time[0]))

analyze.characterization(time, flux, period, star)

```

Listing 17: Función “main” del *script* main.py. Esta función agrupa todas las funciones ya descritas, y las ejecuta, preguntando al usuario si desea simular unos datos o si por el contrario prefiere usar unos datos reales.



## Índice de Figuras

1.1	Curva de luz observada mientras un sistema masivo se interpone ente el observador y la estrella estudiada. Se aprecia como la intensidad de la luz aumenta y disminuye debido al efecto de <i>microlensing</i> . Marcado en azul se ve la irregularidad que produce el exoplaneta al focalizar la luz de la estrella ( <i>5 ways to find a planet</i> ).[1]	2
1.2	Imagen que muestra el concepto de “tránsito”. Se puede apreciar que según el planeta atraviesa la superficie de la estrella, el brillo de la misma disminuye. Cuantizando el descenso, se puede determinar el ratio entre el radio del planeta y el radio de la estrella ( <i>5 ways to find a planet</i> ).[1]	4
1.3	Curva de luz de la estrella Kepler-228. Se observa claramente que es una curva de luz extremadamente ruidosa, lo cual implica una gran necesidad de limpiar los datos. Debido al ruido de la señal recibida, en este caso no se pueden apreciar los tránsitos, al menos a simple vista.	8
1.4	Curva de luz de la estrella Kepler-47. Se aprecia claramente que la mayoría de puntos forman una curva sinusoidal fuera del tránsito debido a un sistema de binarias eclipsantes (Orosz et al., 2019).[9].	9
1.5	Representación artística del satélite Kepler ( <i>Reborn Kepler Can Still Find Planets</i> ).	9
1.6	Plano focal del satélite con los 42 CCDs colocados en 21 de los 25 módulos se empleaban para realizar observaciones. Los otros cuatro módulos se utilizaban para guiar el satélite ( <i>Kepler focal plane assembly</i> ).	10
1.7	Esquema del funcionamiento del filtro de mediana para el filtrado de imágenes. Para filtrar datos el proceso es el mismo, pero en lugar de trabajar con matrices de datos se hace con vectores (Chávez-Burbano, Sarmiento y Ávila, 2009).[11]	12
2.1	Tránsito puro simulado. Se observan disminuciones periódicas de la señal a lo largo de los 50 días simulados en 2.1(a). En 2.1(b) se puede observar los tránsitos con mayor claridad.	15

2.2	Datos simulados. Resultado obtenido al sumar el tránsito puro junto a la variabilidad de la curva de luz debida a una función seno, el <i>offset</i> y el ruido. . . . .	16
2.3	Curva de luz de la estrella simulada una vez se le ha eliminado la mayor parte del ruido, mediante el uso de un filtro de mediana, y la variabilidad. Se puede apreciar su parecido con la figura 2.1. En 2.3(b) se aprecian los tránsitos similares a los de la figura 2.1(b). . . . .	19
2.4	Comparación de la señal limpia y la señal original. Se aprecia que el periodo del tránsito es el mismo, al igual que una disminución de la profundidad del tránsito debido al uso del filtro de mediana. Esto se debe a que el filtro de mediana elimina los detalles más finos. . . . .	19
3.1	En 3.1(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.1(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino. . . . .	24
3.2	Representación del ajuste de la variabilidad sobre los datos simulados una vez pasados por el filtro de mediana. . . . .	25
3.3	Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia $f_{\text{dom}} = 0.4 \text{ días}^{-1}$ , lo cual se corresponde con un periodo orbital $P = f_{\text{dom}}^{-1} = 2.5 \text{ días}$ . . . . .	25
3.4	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la amplitud de la variabilidad. El eje de ordenadas (eje $Y$ ) se muestra en escala logarítmica. . . . .	26
3.5	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la fase de la variabilidad. El eje de ordenadas (eje $Y$ ) se muestra en escala logarítmica. . . . .	27
3.6	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la profundidad del tránsito. El eje de ordenadas (eje $Y$ ) se muestra en escala logarítmica. . . . .	27
3.7	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación de la cantidad de ruido. El eje de ordenadas (eje $Y$ ) se muestra en escala logarítmica. . . . .	28
3.8	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación del periodo del tránsito. El eje de ordenadas (eje $Y$ ) se muestra en escala logarítmica. . . . .	28
3.9	Variación del error relativo a la hora de calcular el periodo del tránsito planetario frente a la variación del periodo de la variabilidad estelar. . . . .	29
3.10	Datos correspondientes a la curva de luz de la estrella TrES-2, obtenidos del catálogo <i>Kepler Objects of Interest</i> . En este caso, los tránsitos planetarios están fuertemente marcados. . . . .	30
3.11	En 3.11(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.11(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino. . . . .	30

3.12 Señal filtrada con la curva de luz que mejor se ajusta a la variabilidad de la estrella. En este caso se aprecia que se trata de una oscilación con amplitud nula. . . . .	31
3.13 Curva de luz de la estrella una vez se le ha eliminado la mayor parte del ruido. . . . .	31
3.14 Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia $f_{\text{dom}} \approx 0.4125 \text{ días}^{-1}$ , lo cual se corresponde con un periodo orbital $P = f_{\text{dom}}^{-1} = 2.4241 \text{ días}$ . . . . .	31
3.15 Datos correspondientes a la curva de luz de la estrella Kepler-75, obtenidos del catálogo <i>Kepler Objects of Interest</i> . . . . .	32
3.16 En 3.16(a) se muestran los datos simulados pasados por el filtro de mediana. En 3.16(b) se muestra la comparación entre los datos simulados originales y los filtrados. Se observa claramente la transformación de la nube de puntos en un conjunto a lo largo de un trazo más fino. . . . .	33
3.17 Señal filtrada con la curva de luz que mejor se ajusta a la variabilidad de la estrella. En este caso se aprecia que la variabilidad no siguen la forma sinusoidal que se esperaba. Esto ha sido una fuente de problemas. . . . .	33
3.18 Curva de luz de la estrella una vez se le ha eliminado la mayor parte del ruido. Se ha intentado eliminar también la variabilidad de la estrella, aunque no se ha tenido mucho éxito en esto último. . . . .	34
3.19 Periodograma obtenido al realizar el espectro de potencias. Se puede observar que el valor máximo del espectro de potencias se corresponde con una frecuencia $f_{\text{dom}} \approx 0.0654 \text{ días}^{-1}$ , lo cual se corresponde con un periodo orbital $P = f_{\text{dom}}^{-1} = 15.3 \text{ días}$ . Como se indica en el texto, este valor tiene una discrepancia del 77% con respecto al valor teórico. . . . .	34
3.20 Periodogramas obtenidos, mediante el uso del <i>software</i> Period04, sin limpiar los datos, 3.20(a), y limpiando los datos, 3.20(b), para los datos simulados. . . . .	35
3.21 Periodogramas obtenidos, mediante el uso del <i>software</i> Period04, sin limpiar los datos, 3.21(a), y limpiando los datos, 3.21(b), para el exoplaneta TrES-2 b. . . . .	35
3.22 Periodogramas obtenidos, mediante el uso del <i>software</i> Period04, sin limpiar los datos, 3.22(a), y limpiando los datos, 3.22(b), para el exoplaneta Kepler-75 b. . . . .	36



## Índice de Tablas

---

3.1	Parámetros del tránsito planetario simulado. . . . .	23
3.2	Parámetros del ruido, de la variabilidad estelar y de la estrella. Se ha escogido $\nu = 3$ para que $f_{\text{NCT}}$ sea perfectamente distinguible de $f_{\text{ND}}$ . . . . .	24
3.3	Parámetros del exoplaneta simulado con los datos de las Tablas 3.1 y 3.2. . . . .	26
3.4	Parámetros estelares de la estrella TrES-2. . . . .	32
3.5	Parámetros del exoplaneta TrES-2 b para este trabajo. Los resultados se presentan junto a su incertidumbre, el valor bibliográfico y la discrepancia. . . . .	32
3.6	Parámetros estelares de la estrella Kepler-75. . . . .	34
3.7	Parámetros del exoplaneta Kepler-75 b para este trabajo. Los resultados se presentan junto a su incertidumbre, el valor bibliográfico y la discrepancia. Se observa que el valor obtenido para el periodo dista del valor bibliográfico. no obstante, en caso de seleccionar el segundo pico del periodograma en lugar de este primero, se obtiene un periodo $P \approx 10.84$ días, compatible con el valor bibliográfico. . . . .	35
3.8	Frecuencias obtenidas por el <i>software</i> Period04 ordenadas según su dominancia en el periodograma de la figura 3.22(a), junto a los periodos correspondientes. . . . .	36

