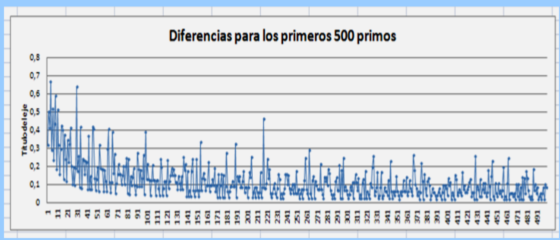
Conjeturas



Edición 2023 Segunda

Colección Hojamat.es

http://www.hojamat.es

PRESENTACIÓN

Este documento contiene comprobaciones de conjeturas efectuadas con hoja de cálculo. El objetivo es puramente didáctico, ya que no se usa una herramienta adecuada, además del hecho de que las conjeturas se intentan demostrar, pero, por su propia naturaleza, nunca se dan por comprobadas. Así que este planteamiento puede resultar engañoso. Lo que se pretende en realidad es suministrar imágenes y esquemas de cálculo que ayuden a interpretar mejor el sentido de cada conjetura.

En esta edición se han añadido comprobaciones con nuestro Buscador de Naturales (descargable desde http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#buscador), por ser una herramienta de fácil uso, que se puede adaptar a las enseñanzas medias o a estudios no reglados. Esta hoja de cálculo no es profesional, por lo que puede bloquearse o no dar los resultados apetecidos. En ese caso, su sencillez permite que se cierre Excel o Calc y se vuelva a comenzar.

TABLA DE CONTENIDO

ANDRICA

Conjetura de Andrica

La conjetura de Andrica se expresa algebraicamente mejor que con palabras. Si representamos por p_n el número primo que aparece en el lugar **n** de su lista, la conjetura se expresa como

$$\sqrt{p_{n+1}} - \sqrt{p_n} < 1$$

"La diferencia entre las raíces cuadradas de dos números primos consecutivos es siempre menor que 1"

Sobre su historia, autor y algunas consideraciones interesantes, en lugar de copiarlas aquí remitimos a una destacada entrada del blog "Gaussianos" (http://gaussianos.com/la-conjetura-de-andrica-o-que-distancia-hay-entre-dos-numeros-primos-consecutivos/)

Lo que nos interesa aquí tiene carácter más humilde, y es la comprobación de esta conjetura con una hoja de cálculo y nivel medio de dificultad. Para ello necesitas dos funciones: ESPRIMO, que te devuelve si un número es primo o no y PRIMPROX, que encuentra el menor número primo que es mayor que uno dado (sea primo o no). Para evitarte tratar con definiciones de funciones y con el BASIC de las hojas, hemos creado la herramienta ***conjeturas.xlsm***, que se encuentra en la dirección

(ver http://hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#conjeturas).

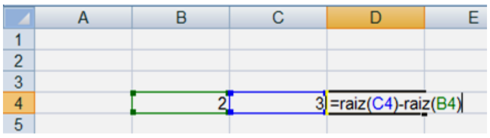
La primera hoja contiene el espacio de trabajo, la segunda el catálogo de funciones implementadas y la tercera los enunciados de las conjeturas. Este archivo se podrá ir actualizando sin previo aviso conforme se vayan tratando conjeturas nuevas.

Supondremos, pues, que tienes abierta la hoja ***conjeturas.xlsm***. Puedes comenzar una tabla en la que figuren en la primera columna todos los números primos (verás cómo) y en la segunda los siguientes primos de cada uno de ellos. Después, en una tercera escribimos la diferencia de las raíces cuadradas de ambos.

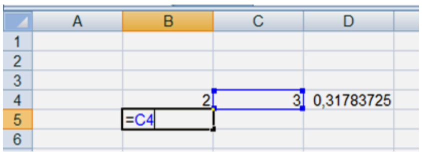
Construcción de la tabla

Comienza, por ejemplo, escribiendo un 2 en la celda B2. Usa la función PRIMPROX para escribir el siguiente primo en C4: =PRIMPROX(B4). Evidentemente obtendrás un 3.

En la celda D4 escribe la diferencia de raíces cuadradas =RAIZ(C4)-RAIZ(B4)



Para que puedas extender la tabla hacia abajo, en la celda B5 copia el contenido de la C4, pero como fórmula, **=C4**. No uses *Copiar y Pegar*. Obtendrás un 3, como era de esperar.



P(n)	P(n+1)	Diferencia raíces
2	3	0,317837245
3	5	0,50401717
5	7	0,409683334
7	**11**	**0,670873479**
11	13	0,288926485
13	17	0,51755435
17	19	0,235793318
19	23	0,43693258
23	29	0,589333284
29	31	0,182599556
31	37	0,514998167
37	41	0,320361707
41	43	0,154314287
43	47	0,298216076
47	53	0,424455289
53	59	0,401035859
59	61	0,129103928
61	67	0,375103096
67	71	0,240797001
71	73	0,117853972
73	79	0,344190672
79	83	0,222239162
83	89	0,323547553
89	97	0,41487667
97	101	0,201017819
101	103	0,099015944

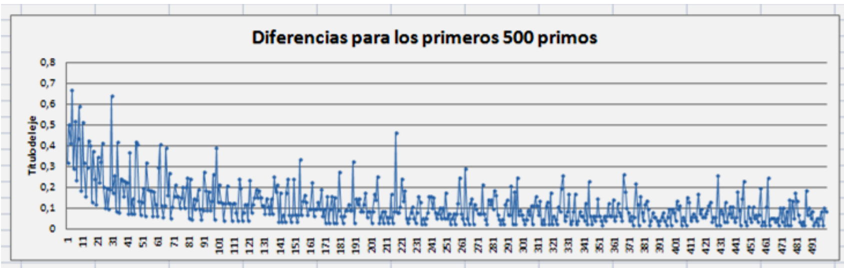
Con el controlador de relleno copia hacia abajo las celdas C4 y D4

P(n)	P(n+1)	Diferencia raíces
2	3	0,317837245
3	5	0,50401717

Lo que te queda por hacer es muy sencillo: de nuevo con el control de relleno copia las tres nuevas celdas de la fila 5 hacia abajo hasta el número de filas que desees:

Hemos marcado en negrita la máxima diferencia, y como era de esperar, todas son menores que la unidad.

Aunque ya están publicados, te puedes dar la satisfacción de crear tu propio gráfico, añadiendo, por ejemplo, otra columna con los números de orden:



En el gráfico se aprecia la máxima diferencia antes de llegar al 11 y que la tendencia general es que, con grandes oscilaciones, los valores tienden a cero, lo que da confianza en que la conjetura sea cierta.

Uso del Buscador

Para una comprobación directa, sin necesidad de manipular la hoja de cálculo, podemos usar estas dos condiciones:

PRIMO
EVALUAR RAIZ(PRIMPROX(N))-RAIZ(N)

La primera busca primos, la segunda pide evaluar la diferencia de raíces entre dos primos consecutivos. Se han buscado entre 1000 y 1050 como una decisión aleatoria:

Núm.	Solución	Detalles
1	1009	6,29E-02
2	1013	9,41E-02
3	1019	3,13E-02
4	1021	0,156098099
5	1031	3,11E-02
6	1033	9,32E-02
7	1039	0,154746559
8	1049	0,030860673

Observamos que todas las diferencias son menores que la unidad.

Otra interpretación

Si representamos por D_n la diferencia entre dos primos consecutivos

$$D_n = P_{n+1} - P_n$$

Si la conjetura es cierta se cumple

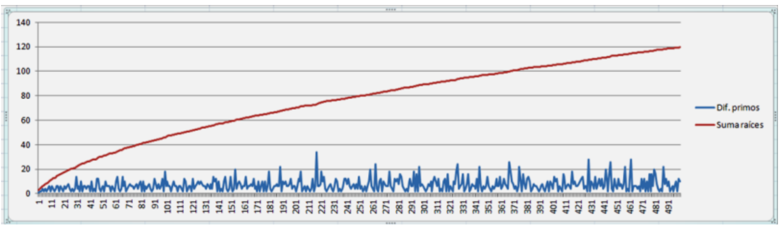
$$D_n = \left(\sqrt{P_{n+1}} + \sqrt{P_n}\right)\left(\sqrt{P_{n+1}} - \sqrt{P_n}\right) < \sqrt{P_{n+1}} + \sqrt{P_n}$$

La diferencia entre dos primos consecutivos siempre es menor que la suma de las raíces cuadradas de ambos.

Es fácil deducir otra expresión más simple:

$$D_n < 2\sqrt{P_n} + 1$$

Puedes crear dos columnas nuevas en tu tabla, una con la suma de raíces y otra con la diferencia de primos consecutivos. Intenta crear un gráfico similar a este:



Contrasta la "suavidad" de la gráfica de la suma de raíces con la de la diferencia de primos. Hay que tener en cuenta que en la primera cada primo se suma en dos datos consecutivos, lo que produce un efecto de promedio, que oculta algo las irregularidades. Lo importante en este caso es se cumple la desigualdad deducida de la conjetura de Andrica.

Uso del Buscador

La nueva desigualdad de Andrica tiene también traducción al Buscador.

Condiciones

PRIMO
EVALUAR 2*RAIZ(N)+1-(PRIMPROX(N)-N)

Restamos ambos miembros de ella para comprobar que la diferencia es positiva. Según la gráfica anterior, posee un crecimiento apreciable. La hemos comprobado entre 20000 y 20100.

Resultado

Solución	Detalles
20011	2,74E+02
20021	2,82E+02
20023	2,78E+02
20029	266,0476992
20047	2,80E+02
20051	2,72E+02
20063	276,2878395
20071	266,3443135
20089	272,4713389

Una interesante generalización

Si la conjetura de Andrica es cierta, podemos plantear la ecuación

$$p_{n+1}^x - p_n^x = 1$$

Tendremos la seguridad de que x estará entre los valores 0,5 y 1. Para cada par de primos consecutivos x tendrá un valor distinto. El máximo lo alcanza para el par (2,3) en el que x=1 y el mínimo en p_{n+1}=127 y p_n=113 con x=0.567148... Este valor es conocido como la constante de Smarandache. La tienes en http://oeis.org/A038458

Es muy instructivo el procedimiento que podemos usar para encontrar el valor de x correspondiente a cada par de números primos consecutivos. Podemos usar para ello la herramienta de **Búsqueda de Objetivos** (lo desarrollamos para Excel, pero es muy fácil trasladarlo a otras hojas)

Tal como se explicó en párrafos anteriores, comienza por crear una tabla de pares de números primos consecutivos. Si te da pereza, usa lo que sigue para un solo par.

P(n)	p(n+1)	x	P(n+1)^x-P(n)^x
2	3	1	1
3	5	1	2
5	7	1	2
7	11	1	4
11	13	1	2
13	17	1	4
17	19	1	2
19	23	1	4
23	29	1	6
29	31	1	2
31	37	1	6
37	41	1	4
41	43	1	2

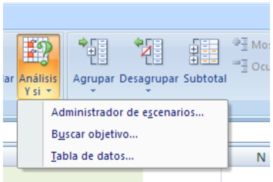
En la tabla hemos añadido una columna para x en la que iniciamos con el valor 1. Una cuarta columna la rellenamos con la fórmula p(n+1)^x-p(n)^x. Si la reproduces, comprueba que los valores que obtienes son los que figuran en la imagen.

Búsqueda del valor de x

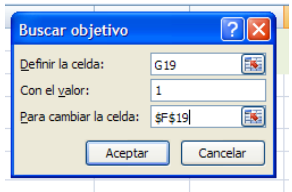
Usaremos la Búsqueda de objetivos para resolver la ecuación

$$p_{n+1}^{x} - p_{n}^{x} = 1$$

Elige un par cualquiera, por ejemplo 29 y 31. Señala la celda que contiene el valor 2 para la diferencia de potencias, y busca el procedimiento ***Buscar Objetivo*** en la fichas **Datos** y grupo **Análisis Y si…**



Ahora, en *Definir la celda* escribes la que contiene la diferencia 2, como *valor* escribes 1, porque ese es tu objetivo, y en *Para cambiar la celda* escribes la celda donde está el valor 1 de la x.



Al pulsar aceptar obtendrás la solución, tal como ves en la imagen:

29	31	0,84555613	1,000125408

La solución, 0,84555… está entre 0,5 y 1, tal como habíamos conjeturado.

Toma el par 113 y 127 y obtendrás la la constante de Smarandache con cinco decimales correctos:

113	127	0,567149642	1,000009903

El problema está en que has de ver cada par uno a uno, pero para un cálculo conjunto nos tendríamos que complicar el proceso.

Puedes consultar más generalizaciones en

http://arxiv.org/ftp/arxiv/papers/0707/0707.2584.pdf

CONJETURA DE LEGENDRE

Esta conjetura afirma que entre dos cuadrados consecutivos n^2 y $(n+1)^2$ existe siempre un número primo.

Se considera básica e importante, por lo que se incluyó en los Problemas de Landau

(http://en.wikipedia.org/wiki/Landau%27s_problems)

Al igual que en la conjetura de Andrica, sólo necesitamos para estudiarla las funciones ESPRIMO y PRIMPROX, incluidas en la herramienta que hemos preparado para el estudio de conjeturas.

(ver http://hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#conjeturas)

Es fácil organizar los cálculos. Diseñamos una columna con los primeros números naturales y junto a ella la de sus cuadrados. Después, a la derecha de cada cuadrado calculamos la función PRIMPROX sobre él para encontrar su próximo primo. Este deberá pertenecer al intervalo formado **por ese cuadrado y el siguiente**:

N	N^2	PRIMPROX(N^2)
1	1	2
2	4	5
3	9	11
4	16	17
5	25	29
6	36	37
7	49	53
8	64	67
9	81	83
10	100	101
11	121	127
12	144	149
13	169	173

A simple vista vemos que cada primo de la tercera columna es menor que el siguiente cuadrado: 67 menor que 81, luego está comprendido entre 64 y 81, o 149, que pertenece al intervalo (144, 169), y así con todos.

Nada impide que comiences la lista no con el 1, sino con un cuadrado mayor, como ves en la imagen

N	N^2	PRIMPROX(N^2)
3201	10246401	10246403
3202	10252804	10252817
3203	10259209	10259237
3204	10265616	10265617
3205	10272025	10272043
3206	10278436	10278461
3207	10284849	10284877
3208	10291264	10291277

Si lo vas a explicar a otras personas, podías añadir una cuarta columna con una fórmula de tipo condicional =SI(el primo es menor que el siguiente cuadrado;”Vale”;”Error”)

3201	10246401	10246403	VALE
3202	10252804	10252817	VALE
3203	10259209	10259237	VALE
3204	10265616	10265617	VALE
3205	10272025	10272043	VALE
3206	10278436	10278461	VALE
3207	10284849	10284877	VALE
3208	10291264	10291277	VALE

De hecho, no existe sólo un número primo entre dos cuadrados, sino que pueden entrar más. Tienes ese dato en http://oeis.org/A014085. Puedes descubrirlo tú con la función PRIMPROX. Sólo copiamos un esquema para el cuadrado de 26, con un resultado de 7 primos:

Número	26
Cuadrado	676
Siguiente cuadrado	729
Primos	
677	
683	
691	
701	
709	
719	
727	

Si construyes bien un esquema similar podrás encontrar el número de primos entre otros cuadrados consecutivos.

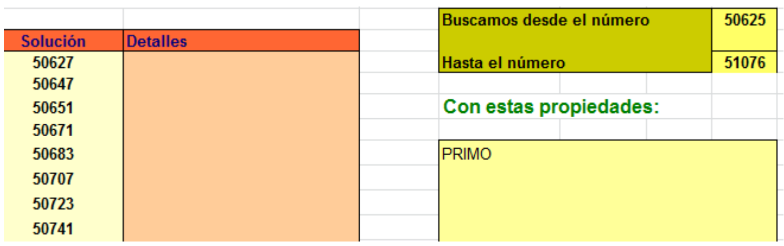
Otro ejercicio sencillo sería, dado un número primo encontrar entre qué cuadrados está. No necesitas saber mucho ¿Cómo se haría? Recuerda la función ENTERO. Ahí tienes un ejemplo:

Primo	288179
536	287296
537	288369

Uso del Buscador

Esta conjetura se comprueba fácilmente con el Buscador. Basta tomar como límites de búsqueda n^2 y $(n+1)^2$ y pedir primos entre ellos. Siempre aparecerá uno al menos:

En la imagen hemos descubierto varios primos entre 225^2 y 226^2:



Otra formulación

Si usamos la función π, que da la distribución de los números primos (π(200) equivaldría a los primos que existen menores o iguales a 200), la conjetura de Legendre se podría expresar así:

$$\pi((n+1)^2) - \pi(n^2) > 0$$

En nuestra herramienta ***conjeturas.xlsm*** hemos implementado la función PPI(n) (le añadimos una p para que no se confunda con el número π, que se expresa como PI()) Con ella es fácil verificar la conjetura: escribes los dos cuadrados consecutivos y le aplicas la función PPI a cada uno. Restas y deberá darte un número mayor que 0. Puedes construirte un esquema de cálculo similar al de la imagen:

	Número	Cuadrado	Función PI
n	217	47089	4857
n+1	218	47524	4899
		Diferencia	**42**

En la página http://oeis.org/A014085 citada más arriba se incluye una generalización de esta conjetura, en el sentido de el exponente 2 se podría sustituir por otro más pequeño. Se ha conjeturado que se podría llegar hasta log(127)/log(16)= 1,74717117169. Se entiende que con carácter general, para todos los valores. Más abajo verás que en un caso particular se puede llegar a valores más pequeños.

Si el esquema anterior lo modificamos para que en lugar de un cuadrado usemos el exponente que deseemos nos servirá para acercarnos al valor mínimo en el que la conjetura sigue siendo cierta:

	Exponente	**1,21**	
	Número	Potencia	Función PI
n	217	671,6086453	121
n+1	218	675,3553697	122
		Diferencia	**1**

Nos hemos dedicado a aproximar este caso al valor mínimo posible y hemos llegado hasta el exponente 1,20545 como mero entretenimiento.

Andrica y Legendre

Si la conjetura de Andrica es cierta, de ella se deduce la de Legendre. En efecto, vimos anteriormente que la diferencia entre un primo P_n y el siguiente P_{n+1}, si la conjetura de Andrica se verdadera, debería cumplir la desigualdad

$$D_n < 2\sqrt{P_n} + 1$$

De ella se deduciría la de Legendre fácilmente. Supongamos que alguien descubre que entre dos cuadrados consecutivos n^2 y $(n+1)^2$ no existe ningún número primo. En este caso llamemos p_n al primo inmediatamente menor que n^2. Sería $p_n<n^2<p_{n+1}$. Según la desigualdad anterior ocurriría que si no existiera ningún primo entre n^2 y $(n+1)^2$ tendríamos

$$D_n = P_{n+1} - P_n > (n+1)^2 - n^2 = 2n + 1 > 2\sqrt{P_n} + 1$$

Esto está en contradicción con la desigualdad previa, luego ha de existir un primo entre ambos cuadrados.

Otra formulación más

Es evidente que la conjetura de Legendre es equivalente a la afirmación de que entre dos números consecutivos n y n+1 siempre existe un número que es la raíz cuadrada de un número primo. Pero esto nos va a dar juego a continuación.

PRIMO MÍNIMO DETRÁS DE UN CUADRADO

En el apartado anterior estudiamos la conjetura de Legendre y terminamos con la formulación siguiente: *La conjetura de Legendre es equivalente a la afirmación de que entre dos números consecutivos n y n+1 siempre existe un número que es la raíz cuadrada de un número primo.*

$$n < \sqrt{p} < n + 1$$

La conjetura de Legendre nos afirma que existe uno al menos, pero lo normal es que existan más. Nos fijaremos en el primer número primo que es mayor que un cuadrado dado n^2, y que, por tanto, su raíz cuadrada sea la más cercana de este tipo al valor de n. Esos valores son fáciles de encontrar. Aquí tienes una función en BASIC:

```
Function primomincuad(n)
 a=n*n+1
while not esprimo(a)
a=a+1
wend
primomincuad=a
End function
```

Dado un valor de n, esa función encuentra el menor número primo que es mayor que su cuadrado. Ya se conocen los valores de estos primos:

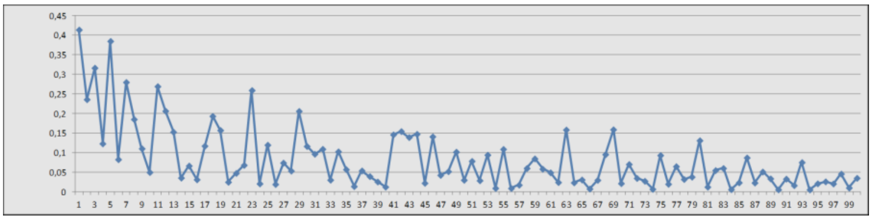
2, 5, 11, 17, 29, 37, 53, 67, 83, 101, 127, 149, 173, 197, 227, 257, 293, 331, 367, 401, 443, 487, 541, 577, 631, 677, 733, 787, 853, 907, 967, 1031, 1091, 1163…

http://oeis.org/A007491

Las raíces cuadradas de estos números estarán comprendidas entre n y n+1. Por ejemplo, el octavo, que es 67, tiene su raíz cuadrada entre 8 y 9, como se ve sin calcularla.

Estos valores nos plantean una pregunta inocente: *¿Qué diferencias concretas existen entre cada número natural y la raíz cuadrada de primo más próxima?*

Para encontrar esa raíz podemos usar la fórmula a =RAIZ(PRIMPROX(N ^ 2)). La hemos utilizado para crear este gráfico de diferencias:



Es muy curioso, porque esas diferencias oscilan con tendencia decreciente desde 0,4142 hasta acercarse a cero. Podíamos plantearlo como una conjetura:

Conjetura 1: Las diferencia entre cualquier número natural y la raíz cuadrada del mínimo número primo mayor que su cuadrado es siempre igual o menor que la raíz cuadrada de 2 menos 1.

Es razonable pensar en esta conjetura. Por una parte la hemos comprobado hasta 5*10^7 con este código PARI:

{for(i=1,5*10^7,b=sqrt(nextprime(i*i))-i;c=sqrt(2)-1;if(b>=c,print(i)))}

Si lo ejecutas verás que sólo imprime el valor 1. Los siguientes números producen diferencias más pequeñas.

Por otra, vemos que los valores son claramente decrecientes en conjunto. Realizamos algunas aproximaciones. ¿Cuántos primos se pueden esperar entre n^2 y $(n+1)^2$?

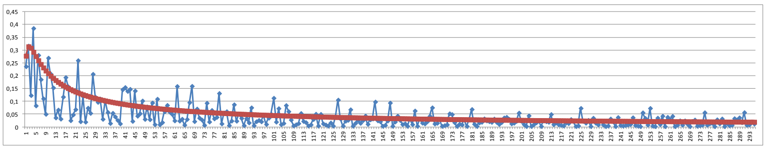
Si usamos el Teorema de los números primos podemos establecer una aproximación grosera

(http://es.wikipedia.org/wiki/Teorema_del_n%C3%BAmero_primo)

$$P = \pi((n+1)^2) - \pi(n^2) \approx \frac{(n+1)^2}{Ln((n+1)^2)} - \frac{n^2}{Ln(n^2)} \approx \frac{2n+1}{2Ln(n)}$$

Y más grosera y atrevida aún: si se esperan P primos entre los dos cuadrados consecutivos, el primero de ellos distará de n^2 una distancia del orden de la fracción inversa, 2Ln(n)/(2n+1). ¿Será así? Recuerda que hablamos de tendencias, no de valores individuales.

Hemos construido una tabla doble: en una columna los valores de RAIZ(PRIMPROX(N ^ 2))-N y en la otra los de 2Ln(n)/(2n+1), con este resultado gráfico:



Vemos que la tendencia decreciente es razonable, luego podemos confiar en que nuestra conjetura sea cierta, que las diferencias nunca son mayores que 0,4142…¡Sólo confiar, nada más!

Conjetura 2: Dado un número natural cualquiera K, existe otro N tal que la diferencia (en valor absoluto) entre su cuadrado N^2 y el mínimo primo mayor que él sea igual a K.

Expresado de otra forma, la expresión PRIMPROX(N ^ 2)-N^2 puede tomar cualquier valor. Esta idea aparece cuando obtienes una lista de valores de N, tomas nota de esa diferencia en una hoja de cálculo y la ordenas después por los valores de la diferencia. No nos cabe aquí la tabla adecuada para que veas que se recorren todos los valores, pero puedes construirla en la hoja de cálculo conjeturas,xlsx

Algunos valores se resisten a salir, como el 29, 68 y 78, pero al final los obtienes. Puedes construir esta función que te devuelve el primer valor posible para K:

```
Public Function difproxprim(k)
Dim n, m
n = 0: m = -1
While k <> m
n = n + 1
m = primprox(n * n) - n * n
Wend
difproxprim = n
End Function
```

Si juegas con ella te darás cuenta de que puede resultar muy lenta en una hoja de cálculo, por ejemplo para obtener que el 68 aparece en la lista para K=5187.

Puedes usar la misma idea en PARI:

```
difproxprim(k)={local(m=-1,n=0);while(k<>m,n=n+1;m=nextprime(n*n)-n*n);return(n)}
{print(difproxprim(68))}
```

En ella sustituyes después el 68 por otro número. Por ejemplo, el 88 se retrasa hasta K=11499 y el 200 hasta K=90963. Es un poco atrevido plantear esta conjetura, pero también es razonable.

CONJETURA DE BROCARD Y OTRAS CUESTIONES

Acabamos de estudiar la conjetura de Legendre

((http://hojaynumeros.blogspot.com.es/2014/04/comprobar-conjeturas-con-hoja-de.html)

Entre dos cuadrados consecutivos n^2 y $(n+1)^2$ existe siempre un número primo.

Se vio también una formulación alternativa:

Si usamos la función π, que da la distribución de los números primos ($\pi(200)$ equivaldría a los primos que existen menores o iguales a 200), la conjetura de Legendre se podría expresar así:

$$\pi((n+1)^2) - \pi(n^2) > 0$$

Lo que no incluimos en esa entrada es que si **n** es un número primo mayor que 2, y estudiamos su cuadrado y el de su siguiente primo, entre ellos no existirá al menos un número primo, sino dos, porque entre los dos cuadrados existirá (salvo el caso de 2 y 3) otro cuadrado intermedio. Resumiendo:

Para n>1, si representamos como p(n) al enésimo número primo, se verificará que entre $p(n)^2$ y $p(n+1)^2$ existirán al menos dos números primos.

Pues bien, Brocard propuso una conjetura más fuerte:

Conjetura de Brocard

*Para n>1, si representamos como p(n) al enésimo número primo, se verificará que entre $p(n)^2$ y $p(n+1)^2$ existirán al menos **cuatro** números primos.*

Podemos construir un modelo de hoja de cálculo para verificar esta conjetura para un número primo cualquiera. Usamos ***conjeturas.xlsm*** como en los casos anteriores.

			Cuadrados		Cuatro primos	
Primo propuesto	2851	Sí es primo	8128201		8128213	
Siguiente primo	2857	Sí es primo	8162449		8128217	
					8128243	
					8128249	

Elegimos un primo (en el ejemplo 2851) y con la función PRIMPROX le encontramos el siguiente debajo (2857). Mediante una fórmula condicional similar a "***=SI(esprimo(F10);"Sí es primo";"No es primo")***" comprobamos que efectivamente ambos son primos. A la derecha les calculamos sus cuadrados.

Para encontrar los cuatro primos comprendidos entre los cuadrados usamos de nuevo PRIMPROX. El primer primo de arriba será el

PRIMPROX del primer cuadrado y los tres restantes serán los próximos primos de los de arriba.

Si el cuarto primo es menor que el segundo cuadrado (8128249<8162449), la conjetura queda comprobada para ese ejemplo. En caso contrario, corre a publicar el contraejemplo, que conseguirás la fama.

Como ocurría con la conjetura de Legendre, en la práctica no sólo existen cuatro primos, sino más. Los tienes publicados en http://oeis.org/A050216. Ahí verás que para n>1 los primos comprendidos son todos mayores que 4: 5, 6, 15, 9, 22, 11, 27, 47, 16, 57, 44, 20, 46, 80, 78, 32, 90, 66, 30, 106,…

Uso del Buscador

Todas estas cuestiones anteriores se podrían haber estudiado con esta herramienta, pero las condiciones son muy parecidas y fáciles de imaginar. En esta otra, al tener que contar primos, sí puede ser interesante su uso. En general, suelen aparecer más de cuatro primos:

Buscamos desde el número	38809		Encontrados
Hasta el número	39601		75
Con estas propiedades:			
PRIMO			Valor de N

En esta captura de pantalla hemos buscado primos entre 197^2 y 199^2, contando 75 de ellos.

Otras posibles situaciones

Nada nos impide plantear cuántos primos existen comprendidos entre dos elementos de cualquier sucesión creciente. Lo hemos estudiado entre cuadrados (Legendre) y entre cuadrados de primos (Brocard). Podíamos verlos entre triangulares consecutivos, por ejemplo. Este caso ya está estudiado y lo puedes consultar en http://oeis.org/A066888

Basta ver la sucesión para entender que se ha conjeturado que siempre existe al menos un número primo entre dos triangulares consecutivos para n>0: 0, 2, 1, 1, 2, 2, 1, 2, 3, 2, 2, 3, 3, 3, 3, 2, 4, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 6, 4,…

Si recuerdas que la fórmula de un número triangular es n(n+1)/2, con ella y el uso de PRIMPROX podrás reproducir este esquema en hoja de cálculo:

Triangulares consecutivos			Primo comprendido
Número de orden	7626		7639
123	7750		

De igual forma se pueden contar los comprendidos entre números oblongos (dobles de triangulares) consecutivos, n(n-1) y n(n+1) Los tienes en http://oeis.org/A108309 y parece lógico conjeturar que siempre existen dos primos entre cada par.

Otras sucesiones se pueden considerar, pero para que tengan interés es conveniente que las diferencias entre cada dos términos consecutivos no crezcan demasiado, lo que facilitaría la presencia de primos intermedios y quitaría interés a la cuestión. Sería el caso, por ejemplo, de las potencias de un número.

Se ha visto la cuestión con semiprimos en http://oeis.org/A088700 y con los términos de la sucesión de Fibonacci (http://oeis.org/A076777) y con seguridad en otros casos que no hemos buscado.

Aquí queremos aportar también nuestra particular sucesión con primos comprendidos. Probamos con los números poderosos

Primos entre poderosos

Llamamos número **poderoso** a aquél en el que todos sus factores primos presentan un exponente mayor que la unidad en la correspondiente descomposición factorial. Son poderosos 1, 4, 8, 9, 16, 25, 27, 32, 36, 49, 64, 72, 81, 100, 108, 121, 125, 128, 144, 169,… http://oeis.org/A001694 En ellos, si un **p** primo divide a **N**, también lo divide su cuadrado, por lo que ninguno de ellos es libre de cuadrados. En virtud de esa definición se ha incluido el 1 en el listado. Por su forma de crecer parecen idóneos para contar primos entre ellos. Lo hemos hecho con este resultado:

4	8	2
8	9	0
9	16	2
16	25	3
25	27	0
27	32	2
32	36	0
36	49	4
49	64	3
64	72	2
72	81	2
81	100	3
100	108	3
108	121	2
121	125	0
125	128	1
128	144	3
144	169	5
169	196	5
196	200	2
200	216	1

Vemos que, por ejemplo, entre 100 y 108 se intercalan tres primos: 101, 103 y 107.

Si escribimos el listado de todas las diferencias observaremos la irregularidad de su distribución

2, 2, 0, 2, 3, 0, 2, 0, 4, 3, 2, 2, 3, 3, 2, 0, 1, 3, 5, 5, 2, 1, 1, 5, 1, 7, 0, 5, 2, 4, 5, 1, 5, 2, 7, 3, 2, 2, 6, 9, 4, 4, 0, 7, 8, 2, 7, 4, 4, 8, 1, 1, 4, 4, 9, 7, 2, 1, 9, 10, 6, 1, 0, 2, 0, 9, 12, 7, 4, 12, 6, 5, 4, 5, 12, 0, 8, 3, 3, 10, 8, 0, 2, 13, 2, 13, 10, 10, 1, 15, 0, 7, 9, 9, 3, 13, …

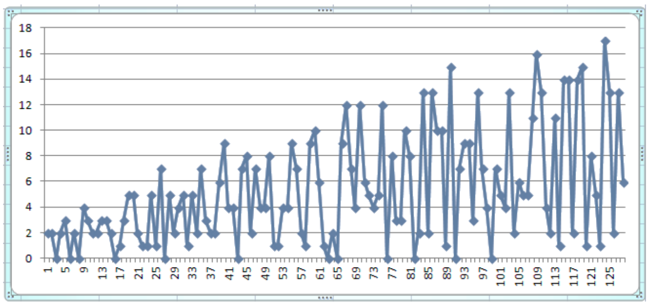
Los puedes buscar con PARI

```
ispowerful(n)={local(h);if(n==1,h=1,h=(vecmin(factor(n)[,2])>1));return(h)}

proxpowerful(n)={local(k);k=n+1;while(!ispowerful(k),k+=1);return(k)}

{for(i=1,5000,if(ispowerful(i),m=proxpowerful(i);p=primepi(m)-primepi(i);print(p)))}
```

No dejan de aparecer ceros, aunque en general las diferencias parecen crecer.



Se asemejan a una vibración que no parara de crecer en amplitud. Como se ve, no hay lugar para una conjetura simple y elegante. Esto es lo normal, no va a resultar una conjetura en cualquier búsqueda que efectuemos.

Hemos publicado esta sucesión en http://oeis.org/A240590

En la parte inferior del gráfico se perciben los puntos de aquellos números poderosos consecutivos que no tienen primos intercalados entre ellos. Son estos:

8, 25, 32, 121, 288, 675, 1331, 1369, 1936, 2187, 2700, 3125, 5324, 6724, 9800, 10800, 12167, 15125, 32761, 39200, 48668… (sólo escribimos el primer elemento del par de poderosos)

	Es primo
1331	FALSO
1332	FALSO
1333	FALSO
1334	FALSO
1335	FALSO
1336	FALSO
1337	FALSO
1338	FALSO
1339	FALSO
1340	FALSO
1341	FALSO
1342	FALSO
1343	FALSO
1344	FALSO
1345	FALSO
1346	FALSO
1347	FALSO
1348	FALSO
1349	FALSO
1350	FALSO
1351	FALSO
1352	FALSO

Por ejemplo, entre el número poderoso 1331 y su siguiente 1352 no existe ni un solo primo.

Esta sucesión permanecía inédita y la hemos publicado en http://oeis.org/A240591

8	1
10	2
14	0
15	2
21	0
22	1
28	1
30	1
33	0
34	0
35	1
38	0
39	1
42	1
46	1
51	1
55	0
57	0

Su carácter creciente justifica que creamos que para un poderoso que no presente ningún primo entre él y el siguiente poderoso, existe otro mayor que él con la misma propiedad. La sucesión tendría infinitos términos.

Compuestos libres de cuadrados

Son números que no son primos y que no tienen divisores cuadrados salvo el 1. Estos dan mejor resultado que los poderosos, en el sentido de que las diferencias no oscilan tanto.

Aquí abundan los ceros y el resto de números presenta máximos que crecen lentamente. Por ejemplo, el primer par que posee tres primos intercalados es 346, que hasta el siguiente compuesto libre de cuadrados, el 354, presenta intercalados los primos 347, 349 y 353. Para llegar a cuatro primos intercalados hay que llegar nada menos que hasta 4584470.

1, 2, 0, 2, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 2, 0, 0, 1, 0, 2, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 2, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 1…

Hemos usado este programa en PARI, además, como hacemos siempre, de una búsqueda previa con hoja de cálculo.

```
freesqrcomp(n)=issquarefree(n)&&!isprime(n)

nextfqc(n)={local(k);k=n+1;while(!freesqrcomp(k),k+=1);return(k)}

primesin(a,b)={local(p=a,q=0);while(p<b,p=nextprime(p);if(p<b,q+=1);p+=1);return(q)}

{for(i=2,100,if(freesqrcomp(i),m=nextfqc(i);p=primesin(i,m);print(i, " ",p)))}
```

Los hemos publicado en http://oeis.org/A240592

También podemos destacar aquí aquellos que no presentan primos en el intervalo respecto a su consecutivo. Son estos:

14, 21, 33, 34, 38, 55, 57, 62, 65, 69, 74, 77, 85, 86, 91, 93, 94, 105, 110, 114, 115, 118, 119, 122, 129, 133, 141, 142, 143, 145, 154, 158, 159, 165, 174, 177, 182, 183, 185, 186, 187, 194, 201, 202, 203, 205, 206, 209, 213, 214, 215,…

Su aparente tendencia a un crecimiento continuado nos hace pensar que la sucesión es indefinida y que siempre existirá otro elemento mayor que uno dado. (http://oeis.org/A240593)

GOLDBACH

La formulación más simple de la Conjetura de Goldbach es:

Todo número par mayor que 2 es suma de dos primos

Fue propuesta por Goldbach el 7 de Junio de 1742, en una carta dirigida a Euler. En realidad, su propuesta se refería a la conjetura ternaria: " *Todo número impar es la suma de tres primos*" y Euler le respondió con la propuesta binaria que todos conocemos.

Ha sido comprobada hasta números muy grandes, pero no se ha podido demostrar. No obstante, se han logrado resultados provisionales:

Cualquier número par es suma de 6 o menos números primos.(Ramaré 1995)

Todo número par suficientemente grande es suma de un primo y del producto de dos primos.(Chen 1966)

Todo número impar N mayor que 5 es suma de tres primos. (Demostración de la conjetura ternaria a cargo de Vinogradov en 1937).

Sólo nos plantearemos, como en toda la serie que vamos desarrollando sobre conjeturas, la comprobación de algunos aspectos de la misma mediante el uso de la hoja de cálculo.

PRIMER NIVEL

Comprobaremos la conjetura en tres niveles distintos, según el uso que se haga del lenguaje de macros. En primer lugar lo efectuaremos con las técnicas usuales de las hojas de cálculo. Usaremos la hoja Conjeturas, alojada en la página

http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#global

(Búscala en la relación de herramientas)

Organizaremos la comprobación según un esquema similar a este:

Conjetura de Goldbach			
Número par mayor que 2			**612**
2			
3			
5	607		
7			
11	601		
13	599		
17			
19	593		
23			

Escribimos un número par mayor que 2 en una celda. En la imagen es el 612. Después ordenamos los números primos en columna, hasta el límite que queramos. Para ello escribimos el 2, debajo **primprox(2)** (función implementada en esta hoja, y que encuentra el primo siguiente a uno dado). Rellenamos hacia abajo y nos resultará la lista de primos.

En una segunda columna escribiremos una fórmula similar a a siguiente, que copiaremos de arriba a abajo:

=SI(Y(F15<=H$11/2;esprimo(H$11-F15));H$11-F15;"")

En ella H11 es la celda donde hemos escrito el 612. En tu caso podrá ser otra. La F15 en nuestro esquema apunta al número primo que tiene a su izquierda. De esa forma, la podemos interpretar así: “Si el primo no llega a la mitad del número par probado (aquí el 612) y su diferencia con él es otro primo, escribo esa diferencia, pero en caso contrario dejo la celda en blanco”.

Es sencillo de entender y funciona escribiendo los pares de primos en los que se descompone el 612. En la imagen 5+607, 11+601, 19+593,…hasta un total de 26 pares. Si no logras ese número, deberás rellenar hacia abajo las dos columnas hasta llegar a la mitad de 612

Este esquema puede aclarar, probando con varios pares, el sentido de la conjetura. También te da confianza en ella, pues no sólo existe un par de primos para cada número par, sino muchos. ¡Pero no se ha probado aún!

NIVEL 2

4	1
6	1
8	1
10	2
12	1
14	2
16	2
18	2
20	2
22	3
24	3
26	3
28	2
30	3

Ya que con el esquema anterior nos han resultado varias descomposiciones en primos para cada número par, podíamos simplificar mucho si lo plasmáramos en una función. En la hoja de cálculo que estamos usando hemos implementado NUMGOLDBACH(N), que devuelve un cero si N no es par y el número de descomposiciones si es par. En el caso del 612 devuelve correctamente 26.

Aquí tienes los primeros resultados. Si la conjetura es cierta, deberán ser todos mayores que 0. Están recogidos en http://oeis.org/A045917

Merece la pena recorrer la codificación de esta función y así entenderás mejor las cuestiones.

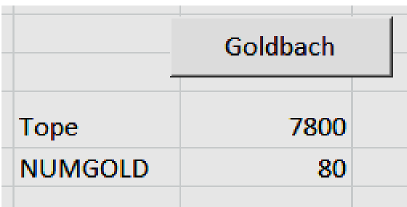
Public Function numgoldbach(n)
Dim ng, i

If n <> 2 * Int(n / 2) Then ‘si es impar devuelve un cero (valor de ng)
ng = 0
Else
i = 2: ng = 0
While i <= n / 2 ‘si es par recorre todas las posibles sumas de primos
If esprimo(n - i) Then ng = ng + 1 ‘si el segundo sumando es primo, incrementa el contador ng
i = primprox(i) ‘esta línea asegura que el primer sumando sea primo
Wend
End If
numgoldbach = ng
End Function

NIVEL 3

Podemos dejar que sea la hoja de cálculo la que recorra automáticamente los primeros números hasta un tope o hasta que ***numgoldbach*** dé un cero. Como lo segundo es imposible para números pequeños (ya está comprobada la conjetura), el resultado final será siempre un cero.

Podíamos usar un esquema similar al siguiente:



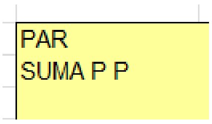
Escribimos un tope, pulsamos el botón e irán apareciendo valores de Numgoldbach, ninguno nulo, hasta finalizar la búsqueda. Si uno fuera cero, se interrumpiría el proceso con un solemne mensaje. La programación del botón podría ser similar a esta:

```
Sub buscagoldbach()
Dim i, g, p

p = ActiveWorkbook.Sheets(1).Cells(5, 3).Value ‘lee el tope
i = 4 ‘inicio búsqueda
g = 1’inicio valor de numgolbach
While g <> 0 And i <= p
i = i + 2 ‘busca de 2 en 2
g = numgoldbach(i)
ActiveWorkbook.Sheets(1).Cells(6, 3).Value = g ‘escribe el valor de g
If g = 0 Then MsgBox ("¡Contraejemplo!") ‘Esto no va a ocurrir
Wend
End Sub
```

Uso del Buscador

Condiciones



Significa que el número ha de ser PAR y suma de dos primos. La segunda es redundante si es cierta la conjetura, pero nos da un modo rápido de observar dos primos que hacen que se cumpla. En la imagen figuran las sumas de los números pares comprendidos entre 500 y 525. Solo nos da una solución. Puedes cambiar a números más grandes para seguir comprobando lo conjetura.

Resultado

Solución	Detalles
500	13 + 487
502	3 + 499
504	5 + 499
506	3 + 503
508	5 + 503
510	7 + 503
512	3 + 509
514	5 + 509
516	7 + 509
518	19 + 499
520	11 + 509
522	13 + 509
524	3 + 521

Variantes

Variante ternaria

"Todo número impar mayor que 5 es la suma de tres primos"

No vamos a repetir con ella los tres niveles anteriores. El primer nivel necesitaría una estructura de datos tridimensional, poco intuitiva en una hoja de dos dimensiones. El tercero sería semejante al del primer caso. Así que sólo desarrollaremos un esquema con todas las posibles descomposiciones en tres sumandos primos:

Escribe el número	**29**		Golbach para impares
	13	11	5
	17	7	5
	19	5	5
	19	7	3
	23	3	3

Como en el caso anterior, no vamos a analizar si el número es impar o no. Simplemente hemos programado un botón que lo descompone en esos sumandos de todas las formas posibles (lo haremos con sumandos decrecientes)

Para quien le guste la programación, ahí tiene explicado el algoritmo que hemos usado:

```
Sub goldbach3()
Dim fila, a, b, c, n

fila = 7
n = ActiveWorkbook.Sheets(1).Cells(fila, 3).Value ‘lee el número, que se encuentra en C7
a = 2 ‘primer sumando primo
While a < n ‘ el primer sumando llega hasta n en lo posible
b = 2
While b < a ‘ el segundo es inferior al primero
c = n - a – b ‘tercer sumando
If esprimo(c) And c <= b Then ‘si el tercer sumando también es primo, se presenta el resultado
fila = fila + 1
ActiveWorkbook.Sheets(1).Cells(fila, 3).Value = a
ActiveWorkbook.Sheets(1).Cells(fila, 4).Value = b
ActiveWorkbook.Sheets(1).Cells(fila, 5).Value = c
End If
b = primprox(b) ‘se incrementa el segundo primo
Wend
a = primprox(a) se incrementa el primero
Wend
End Sub
```

Como era de esperar, siempre aparecen los tres sumandos primos. Se deja a los lectores el definir una función que cuente las soluciones. Siempre existirá al menos una.

Uso del Buscador

Basta estudiar la siguiente captura de pantalla: buscamos entre 40 y 60 los impares y su descomposición en suma de tres primos:

Solución	Detalles
41	2 + 2 + 37
43	3 + 3 + 37
45	2 + 2 + 41
47	2 + 2 + 43
49	3 + 3 + 43
51	2 + 2 + 47
53	3 + 3 + 47
55	3 + 5 + 47
57	2 + 2 + 53
59	3 + 3 + 53

Buscamos desde el número	40
Hasta el número	60

Con estas propiedades:

NO PAR
SUMA P P P

Expresión mediante equidistancia

Un comentario a la entrada http://culturacientifica.com/2013/06/26/la-conjetura-de-goldbach/ me ha dado la idea de organizar una comprobación distinta.

Si la conjetura es cierta, para todo número par 2N, si es la suma de dos primos p y q, con p>q, cumplirán que p+q=2N, o bien que p-N=N-q:

Todo número entero positivo mayor que 4 es equidistante de dos primos

Es fácil ver que es otra formulación distinta de la conjetura de Goldbach. En los párrafos anteriores hemos visto la consecuencia directa. A la inversa, si es cierto que todo N equidista de dos primos, dado un par 2N aplicamos p-N=N-q para cierto par de primos, con lo que 2N=p+q. El exigir que sea mayor que 4 es porque no habría primos inferiores para números menores.

Escribe el número	29			
	1	28	30	
	2	27	31	
	3	26	32	
	4	25	33	
	5	24	34	
	6	23	35	
	7	22	36	
	8	21	37	
	9	20	38	
	10	19	39	
	11	18	40	
	12	17	41	SI
	13	16	42	
	14	15	43	
	15	14	44	

Es muy fácil organizar la comprobación con esta variante. Lo efectuaremos en el Nivel 1, de cálculo manual:

Escribimos la lista de números consecutivos 1, 2, 3,…y los sumamos y restamos con el número dado. Después, en la tercera columna, escribimos una fórmula similar a =SI(Y(ESPRIMO(D9);ESPRIMO(E9));"SI";"") que nos devuelve un “SI” si los equidistantes son primos.

En la imagen, 17=29-12 y 41=29+12.

CONJETURA N^2+1

Es uno de los problemas de Landau, y en el momento de redactar este texto sigue sin conocerse si es verdadera o no la siguiente conjetura:

Existen infinitos primos de la forma n^2+1

Hardy y Littlewood supusieron que la conjetura era verdadera, y aproximaron el número de tales primos menores que n, P (n), asintóticamente a

$$P(n) = C\frac{\sqrt{n}}{\ln(n)}$$

Con C una constante adecuada.

El listado de los primeros primos de este tipo lo puedes consultar en http://oeis.org/A002496

2, 5, 17, 37, 101, 197, 257, 401, 577, 677, 1297, 1601, 2917, 3137, 4357, 5477, 7057, 8101, 8837, 12101, 13457, 14401,…

Si la conjetura es cierta, esta sucesión deberá poseer infinitos términos.

¿Qué estudios podríamos abordar sobre este tema con una hoja de cálculo?

El primer objetivo razonable es el de comprobar que, dado un número cualquiera, existe un número primo del tipo n^2+1 que es mayor que él.

Usaremos la herramienta de hoja de cálculo ***conjeturas***, alojada en

http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#global

Para encontrar ese primo mayor que el dado, reiteraremos el uso de la función PRIMPROX hasta que encontremos un número primo **p** tal que **p-1** sea un cuadrado.

(A) Planteamiento manual

Escribe un número entero positivo			1500
Próximo primo		**¿Es del tipo N^2+1?**	
1511			
1523			
1531			
1543			
1549			
1553			
1559			
1567			
1571			
1579			
1583			
1597			
1601		1601	

Basta estudiar este esquema brevemente para descubrir su funcionamiento:

El primer número primo de la lista es el PRIMPROX(N), en la imagen 1511. Los siguientes se obtienen como los próximos primos del de la fila superior. Esta lista se puede extender hacia abajo todo lo que se desee.

En la segunda columna hemos usado una fórmula del tipo

=SI(ESCUAD(C9-1);C9;""), es decir, si C9 u otro primo de la lista cumple que al restarle la unidad se convierte en un cuadrado, lo escribimos, y , si no, dejamos la celda en blanco. Así descubrimos que el primer primo de este tipo es 1601. Si la conjetura es cierta, siempre llegaremos a un número de ese tipo.

Este método puede necesitar muchas filas hasta dar con el primo esperado. Por eso, se puede plantear como una función:

(B) Estudio mediante una función

Si suponemos cierta la conjetura, para cada número existirá un primo mayor que él con la forma n^2+1. Entonces lo podemos plantear como una función. Su listado lo entenderás fácilmente:

```
Public Function proxn2mas1(n)
Dim p
p = primprox(n)
While Not escuad(p - 1)
p = primprox(p)
Wend
proxn2mas1 = p
End Function
```

De esta forma, la búsqueda manual que emprendimos en el caso anterior la podemos reducir al planteamiento de esta función:

Número N	Próximo primo n^2+1	
1500	1601	

Se comprende que para números grandes esta función tardará algo en calcularse. Lo hemos intentado con 10^7:

Número N	Próximo primo n^2+1	
10000000	10074277	
	3174	

Tarda unos segundos, aunque no es un retraso desesperante. Hemos añadido la raíz cuadrada del primo menos uno, 3174.

Lista de primos de este tipo

Con esa función ***proxn2mas1*** podemos reproducir toda la lista de OEIS. Basta escribir un 2, debajo de él *proxn2mas1(2)* y nos resultará un 5. Le aplicamos de nuevo *proxn2mas1* y obtendremos el 17, y así seguimos hasta donde deseemos.

2
5
17
37
101
197
257
401
577
677
1297
1601
2917
3137
4357
5477
7057

Si en lugar de comenzar con el 2 inicias con un número cualquiera, se escribirá la continuación de la lista, salvo quizás el primero, que no tiene que ser primo de ese tipo. Aquí tienes los siguientes a 10000:

10000
12101
13457
14401
15377
15877
16901
17957
21317
22501
24337
25601
28901
30977
32401
33857

Uso del Buscador para comprobar la conjetura

Condiciones

Buscamos desde el número	10000
Hasta el número	15000
Con estas propiedades:	
PRIMO ES CUADRADO(N-1)	

Buscamos, entre 10000 y 15000 los primos deseados. Resultarán los tres únicos primos de la forma N^2+1.

Resultado

Solución
12101
13457
14401

Podemos elegir unos límites mayores, pero se tardará más tiempo en obtener la solución, Estos son los comprendidos entre 1000000 y 1100000:

Solución
1008017
1020101
1073297

Si al planteamiento le añadimos la condición EVALUAR RAIZ(N-1), nos dará en la columna de Detalles el valor de N en N^2+1:

Solución	Detalles
101	10
197	14
257	16
401	20
577	24
677	26

Aproximación asintótica

Para comprobar la aproximación de Hardy y Littlewood necesitamos contar los primos de este tipo anteriores a N. Algo parecido a la función PRIME(N), pero quedándonos sólo con los primos de forma n^2+1

Entenderás a la primera esta definición:

```
Public Function ppn2mas1(n)
Dim pp, i
' para valores de n superiores s 2

i = 2: pp = 0
While i <= n
pp = pp + 1
i = proxn2mas1(i)
Wend
ppn2mas1 = pp
End Function
```

Esta función cuenta los primos del tipo n^2+1 inferiores o iguales a N. Como nos interesan valores grandes por cuestiones asintóticas, suponemos, para simplificar la programación, que N es mayor que 2. Observa esta tabla en la que se percibe que tratamos con una función escalonada, y que los cambios ocurren en 5 y 17, primos del tipo estudiado.

N	PPN2MAS1(N)
3	1
4	1
5	2
6	2
7	2
8	2
9	2
10	2
11	2
12	2
13	2
14	2
15	2
16	2
17	3
18	3

Para comprobar la aproximación asintótica y evaluar la constante C crearemos una tabla a partir del 10 en progresión geométrica hasta llegar a 10000000:

N	PPN2MAS1(N)	RAIZ(N)/LN(N)	CONSTANTE C
10	2	1,373359738	1,45628268
100	4	2,17147241	1,842068074
1000	10	4,577865794	2,18442402
10000	19	10,85736205	1,749964671
100000	51	27,46719476	1,856760417
1000000	112	72,38241365	1,547337182
10000000	316	196,1942483	1,610648644

Hemos evaluado la constante C como cociente entre la función de distribución de los primos de tipo n^2+1 inferiores a N y la aproximación RAIZ(N)/LN(N). No usamos una herramienta adecuada, pero se ve que los valores de C presentan una cierta convergencia.

Variante de la conjetura

La más sencilla es la que busca primos de la forma n^2+a. Podemos crear una función similar a la que hemos usado, pero añadiendo un parámetro A

```
Public Function proxn2masa(n,a)
Dim p

p = primprox(n)
While Not escuad(p - a)
p = primprox(p)
Wend
proxn2masa = p
End Function
```

Con esta función se puede comprobar que, dado cualquier valor de N (primo o no) y elegida una constante A, existe un número primo del tipo N^2+A superior a N

Observa un posible esquema de búsqueda:

Número N	Valor de A
300000	7
Primo superior del tipo n2+a	
302507	
Primo menos A	
302500	
Raiz	
550	

CONJETURA DE POLIGNAC

Se llama **Conjetura de Polignac** a la enunciada por Alphonse de Polignac in 1849 y que se puede expresar así:

Hay un número infinito de números primos (p, q) tales que p - q = k, siendo k un número par.

Últimamente se ha hablado más de ella por algunos avances que se han producido y que pudieran llevar a su demostración

(Ver http://gaussianos.com/de-70000000-700-en-seis-meses/ y http://en.wikipedia.org/wiki/Polignac%27s_conjecture)

Dentro de esta conjetura, y para k=2 se incluye la de los primos gemelos:

Existen infinitos pares de primos gemelos (p, p+2)

(http://es.wikipedia.org/wiki/Conjetura_de_los_n%C3%BAmeros_primos_gemelos)

Pero también podríamos expresar lo mismo para pares del tipo (p, p+4), “cousin primes” y los del tipo (p, p+6), “sexy primes”.

Nosotros trabajaremos con el enunciado general, sobre los pares (p, p+k), con k número par. Para ello, usaremos el valor de k como entrada a algoritmos de búsqueda.

Esquema sin macros

Para buscar estos pares (p, p+k) ambos primos podemos crear en la hoja Conjeturas

(http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#global) una lista de números primos **p** en forma de columna. Para ello la iniciamos con el número que deseemos, generalmente grande, y después, en cada fila usamos la función PRIMPROX(N), siendo N el número contenido en la celda de arriba, con lo que para aumentar la lista bastará extender la fórmula hacia abajo. Después creamos una columna paralela formada por **p+k**. Por último, en una tercera columna usamos la función ESPRIMO. De esta forma, la aparición del primer par con ambos primos se detectará por el valor

Espacio de trabajo

Organiza aquí tus cálculos y esquemas

Inicio	1000000		
Valor de k	2		
	p	p+k	ESPRIMO
	1000003	1000005	FALSO
	1000033	1000035	FALSO
	1000037	1000039	VERDADERO
	1000039	1000041	FALSO
	1000081	1000083	FALSO
	1000099	1000101	FALSO
	1000117	1000119	FALSO
	1000121	1000123	FALSO
	1000133	1000135	FALSO
	1000151	1000153	FALSO

VERDADERO de esa función.

En la imagen hemos detectado el primer par de números gemelos después del número 1000000, el (1000037,1000039), Se comprende que si la conjetura es verdadera, prolongando las columnas lo que sea necesario, encontraremos un par de primos con la diferencia que hayamos fijado.

Por ejemplo, aquí tienes los primeros primos diferenciados en 1000 que siguen a 10000000:

10000261	10001261	VERDADERO

La columna puede hacerse muy alta, y nos convendría también poder encontrar los pares buscados con una sola función.

Función Polignac

Para abreviar el proceso usaremos la función POLIGNAC(P;K), que hemos añadido a nuestra hoja cálculo ***conjeturas***, enlazada más arriba.

Esta función actúa sobre un inicio **n** y una diferencia par **k**, y encuentra el primer número primo **p** posterior al inicio que forma par de primos con p+k. Basta leer los comentarios para entender su funcionamiento.

Function polignac(n, k)
Dim novale As Boolean
Dim p, q

p = n 'Iniciamos la búsqueda en n
novale = True 'Variable para terminar la búsqueda
If k / 2 <> k \ 2 Then polignac = 0: Exit Function 'Si k es impar, damos valor 0 a la función
While novale

p = primprox(p) ‘Buscamos el siguiente primo
If esprimo(p + k) Then q = p: novale = False ‘Si encontramos (p,p+k) paramos
Wend
polignac = q ‘Damos a la función el valor del primer primo del par
End Function

Por ejemplo, deseamos comprobar la conjetura encontrando el primer par de números primos diferenciados en 2000 que sigue al inicio 10^6. En la imagen, capturada de la hoja conjeturas.xslm tienes la respuesta: 1000121 es el primer primo tal que al sumarle 2000 se obtiene otro primo 1002121. El resto del esquema lo hemos organizado para que se obtengan varios pares en lugar de uno solo. A la derecha hemos incluido la prueba de que ambos son primos. Te invitamos a que construyas un esquema similar en la hoja ***conjeturas***.

Espacio de trabajo				
Organiza aquí tus cálculos y esquemas				
Número inicial	1000000			
Valor de k	2000			
	p	p+k	Son primos	
	1000121	1002121	VERDADERO	VERDADERO
	1000151	1002151	VERDADERO	VERDADERO
	1000289	1002289	VERDADERO	VERDADERO
	1000403	1002403	VERDADERO	VERDADERO
	1000427	1002427	VERDADERO	VERDADERO
	1000457	1002457	VERDADERO	VERDADERO

Como en todas las cuestiones, la hoja de cálculo puede fallar para números muy grandes, por lo que debemos acudir a programas más potentes. Elegimos PARI por ser gratuito. Podemos usar la siguiente función, a la que hemos añadido un “print” para que veas el resultado:

polignac(n,k)={local(p);p=nextprime(n);while(!isprime(p+k),p=nextprime(p+1)); return(p)}

{print(polignac(10^6,2000))}

Si lo ejecutas obtendrás el mismo número primo que con la hoja, 1000121:

```
parisize = 4000000, primelimit = 500000
? \r ini.txt
%1 = (n,k)->local(p);p=nextprime(n);while(!isprime(p*k),p=nextprime(p+1));return
(p)
1000121
?
```

También aquí puede fallar la función para números muy grandes, pero el margen es mayor que en Excel.

La idea es que si la conjetura es cierta (y la herramienta de cálculo lo admite), podemos elegir un número de inicio arbitrariamente grande y siempre tendremos un resultado.

Uso del Buscador

Elegimos un número par K, por ejemplo el 28, y basta buscar lo siguiente:

Condiciones

PRIMO
ES PRIMO(N+28)
EVALUAR N+28

Resultado

En la siguiente tabla figuran las soluciones obtenidas entre 100000 y 101000:

Solución	Detalles
100333	100361
100363	100391
100483	100511
100519	100547
100621	100649
100741	100769
100801	100829
100981	101009
100999	101027

Como en otros casos, se pueden elegir números de más cifras, pero cuidando no superen la capacidad de las hojas de cálculo para tratar con números enteros.

Lista de números primos gemelos, "cousin", "sexy" y otros

Con esta función puedes crearte fácilmente una lista de primos gemelos. Basta usarla de forma recurrente en columna a partir del 1, y con una diferencia dada. En el esquema que adjuntamos más arriba usamos esta técnica:

Inicio	1	
Valor de k	2	
	p	p+k
	3	5
	5	7
	11	13
	17	19
	29	31
	41	43
	59	61
	71	73
	101	103
	107	109

El primer par de primos gemelos (3,5) se ha construido a partir del 1. Los siguientes, a partir del anterior, con la función polignac(p,k). Las columna de la derecha la programamos para que sume el valor de **k** a la de la izquierda.

Si la conjetura de Polignac es cierta, esta tabla tendría una altura infinita. No terminarían de aparecer pares.

Podemos construir de esta forma los pares de primos "cousin", que se diferencian en 4.

Inicio	1	
Valor de k	4	
	p	p+k
	3	7
	7	11
	13	17
	19	23
	37	41
	43	47
	67	71
	79	83
	97	101
	103	107

De igual forma, los “sexy”:

Inicio	1	
Valor de k	6	
	p	p+k
	5	11
	7	13
	11	17
	13	19
	17	23
	23	29
	31	37
	37	43
	41	47
	47	53

Nada nos impide crear una lista personalizada. Por ejemplo, si has nacido en el año 1962, como es par, puedes crear pares de primos con esa diferencia:

Inicio	1	
Valor de k	1962	
	p	p+k
	11	1973
	17	1979
	31	1993
	37	1999
	41	2003
	67	2029
	101	2063
	107	2069
	127	2089
	137	2099

Llama la atención que el primer elemento del par no tiene que ser muy grande aunque k lo sea. Observa la lista de pares con una diferencia de 1000000:

Inicio	1	
Valor de k	1000000	
	p	p+k
	3	1000003
	37	1000037
	151	1000151
	193	1000193
	199	1000199
	211	1000211
	313	1000313
	367	1000367
	397	1000397
	409	1000409

Otras posibilidades

No podemos construir tríos de primos de este tipo, porque siempre uno de ellos sería múltiplo de 3, pero sí tríos de la forma (p, p+2, p+6), o, en general (p, p+k, p+3k). Para explorar un poco por este camino, bastaría sustituir la línea de Basic

If esprimo(p + k) Then q = p: novale = False

Por esta otra

If esprimo(p + k) And esprimo(p + 3 * k) Then q = p: novale = False

Si la conjetrura de Polignac es cierta, estas búsquedas terminarán por dar un resultado. Observa el caso de n=1000000 y k=1000

Inicio	1000000		
Valor de k	2000		
	p	p+k	p+3k
	1000151	1002151	1006151
	1000721	1002721	1006721
	1001381	1003381	1007381
	1001549	1003549	1007549
	1003049	1005049	1009049
	1005359	1007359	1011359
	1005827	1007827	1011827
	1006433	1008433	1012433
	1006547	1008547	1012547
	1007609	1009609	1013609

Casi todos estos casos están publicados y no pasan de ser curiosidades derivadas de la conjetura que estudiamos. Si eliges un ejemplo inadecuado, como (p, p+2, p+4), puede ocurrirte que se bloquee la hoja de cálculo y tengas que recurrir al Administrador de tareas para cerrarla.

Por el contrario, si consigues tríos no publicados, podrías intentar incluirlos en OEIS.

PRIMOS DE FIBONACCI

Hoy estudiaremos otra conjetura bastante popular:

Existen infinitos números de Fibonacci que son primos.

Así que si construimos la sucesión de Fibonacci y elegimos los términos que sean primos, encontraremos uno de ellos que sea mayor que cualquier otro entero que imaginemos. Aprovecharemos esta conjetura para repasar conceptos, construir algoritmos y explicar algunas propiedades de los números de Fibonacci.

Los primeros primos de Fibonacci son estos:

2, 3, 5, 13, 89, 233, 1597, 28657, 514229, 433494437, 2971215073, 99194853094755497,…

(http://oeis.org/A005478)

Según la conjetura, esta sucesión debería tener infinitos términos. No es intuitivo, porque en cada aumento de índice resulta más improbable que el término correspondiente sea primo, pero así son las conjeturas, que se encuentran a veces en el término de separación entre lo imposible e improbable.

Comprobación de la conjetura

En la hoja CONJETURAS

(http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#conjeturas)

dispones de las funciones necesarias para comprobar la conjetura, se entiende que en unos pocos ejemplos. La primera, ESPRIMO, ya ha sido presentada muchas veces en estos documentos (escribe ESPRIMO HOJA en un navegador de Internet), pero necesitamos otra, ESFIBO, que nos indica si un número pertenece o no a la sucesión de Fibonacci. Esta función se basa en un popular criterio para saber si un número es de Fibonacci o no: ***Un número N pertenece a la sucesión de Fibonacci si y sólo si 5N^2+4 o 5N^2-4 es un cuadrado perfecto.***

(Ver http://gaussianos.com/algunas-curiosidades-sobre-los-numeros-de-fibonacci/)

Según eso, ésta puede ser la función que devuelva VERDADERO si un número es del tipo Fibonacci y FALSO en el caso opuesto:

```
Public Function esfibo(n) As Boolean 'devuelve verdadero si N es de Fibonacci
Dim f As Boolean
Dim a

f = False
a = 5 * n * n + 4
If escuad(a) Then f = True
a = 5 * n * n - 4
If escuad(a) Then f = True
esfibo = f
End Function
```

Disponiendo de esta función y de la ESPRIMO, podemos construir otra, a la que llamaremos FIBOPRIMPROX, que, dado un entero positivo, devuelva el menor primo Fibonacci que es mayor que él, es decir, el "próximo primo Fibonacci". Su código sería:

```
Function fiboprimprox(a) As Long
Dim p, prim As Long
Dim sale As Boolean

'Encuentra el menor primo de Fibonacci mayor que el dado
p = a + 1: sale = False: prim = 0
While Not sale
If esprimo(p) And esfibo(p) Then prim = p: sale = True
p = p + 1
Wend
fiboprimprox = prim
End Function
```

Se entiende fácilmente. El problema, como ocurre frecuentemente, es que la hoja de cálculo tiene una capacidad limitada de cálculo con enteros. Por ello, con la hoja CONJETURAS sólo hemos podido llegar al próximo a 100000.

¿Existen infinitos primos de Fibonacci?

Número positivo	Próximo primo Fibonacci
10	13
100	233
1000	1597
10000	28657
100000	514229

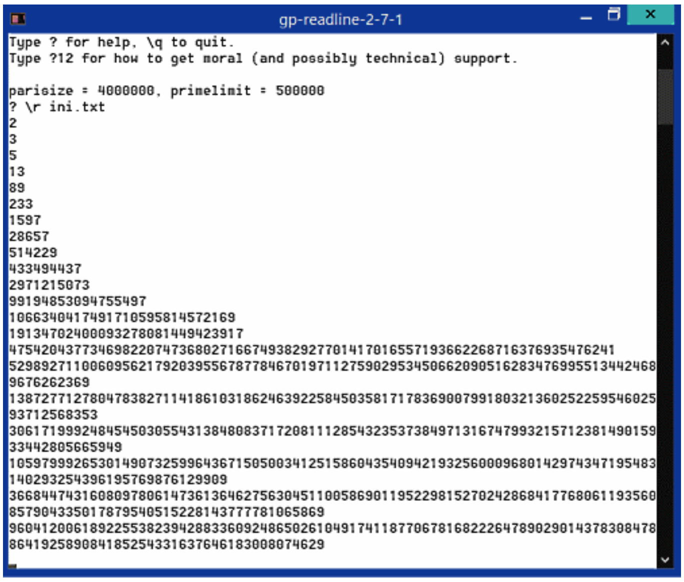
Observa que los números de la segunda columna pertenecen a la sucesión de primos Fibonacci. El siguiente, 433494437, sería difícil de obtener con este procedimiento.

Si la conjetura es cierta, la función FIBOPRIMPROX(N) debe devolver un resultado por muy grande que sea N.

Como procedemos a menudo, traducimos el proceso a PARI para ver si podemos reproducir más elementos de la sucesión. Hemos optado por este código:

{for(i=1,10^4,f=fibonacci(i);if(isprime(f),print(f)))}

El resultado ha sido impresionante, porque en pocos segundos nos ha devuelto los primos contenidos en los 10000 primeros términos de la sucesión de Fibonacci:



Si en el código PARI hubiéramos pedido el valor del índice en lugar del término de Fibonacci nos hubiera resultado la sucesión:

3, 4, 5, 7, 11, 13, 17, 23, 29, 43, 47, 83, 131, 137, 359, 431, 433, 449, 509, 569, 571, 2971, 4723, 5387, 9311, 9677,…

Como se ve, salvo el caso del 4, todos los índices de números de Fibonacci primos son también primos. Esto se deriva de que si p divide a q, Fp también divide a Fq para p,q>=3. Así que si el número de orden no es primo, tampoco lo será el número Fibonacci correspondiente. La propiedad recíproca no es cierta. Por ejemplo, el término de índice 19, primo, es 4181=37*113, compuesto.

Divisibilidad en los números Fibonacci

La cuestión anterior da pie a que revisemos algunas propiedades interesantes que presentan los factores primos de los términos de esta sucesión.

El máximo común divisor

Los términos de la sucesión de Fibonacci cumplen la siguiente curiosa propiedad

$$MCD(F_m, F_n) = F_{MCD(m,n)}$$

Por ejemplo, si elegimos los términos 24 y 36 de la sucesión,

F(24) = 46368 $=2^5*3^2*7*23$ y F(36) = 14930352 = $2^4*3^3*17*19*107$, tendremos

MCD(46368, 14930352) = 144, MCD(24, 36) = 12 y F(12)=144

Por tanto, si el índice de un número de Fibonacci es primo, éste será coprimo con el anterior y el siguiente. Obviamente, esto lo cumplirán los primos Fibonacci, pero también el contraejemplo que vimos más arriba, F(19) = 4181. En la tabla verás la falta de elementos comunes con el anterior y el siguiente elemento Fibonacci.

N	F(N)	Factores
18	2584	[2,3][17,1][19,1]
19	4181	[37,1][113,1]
20	6765	[3,1][5,1][11,1][41,1]

Teorema de Carmichael

Relacionado con este tema de divisores de los números Fibonacci disponemos de este interesante teorema:

Todo término de la sucesión de Fibonacci distinto de 1, 8 y 144, posee un factor primo que no divide al anterior término.

Lo puedes comprobar en esta tabla de divisores de los términos 10 a 20:

N	F(N)	Factores
10	55	[5,1][11,1]
11	89	[89,1]
12	144	[2,4][3,2]
13	233	[233,1]
14	377	[13,1][29,1]
15	610	[2,1][5,1][61,1]
16	987	[3,1][7,1][47,1]
17	1597	[1597,1]
18	2584	[2,3][17,1][19,1]
19	4181	[37,1][113,1]
20	6765	[3,1][5,1][11,1][41,1]

Todo término posee un factor primo que no divide al anterior (recuerda que el segundo número de cada corchete es el exponente del factor primo).

Puede ocurrir que un factor dado no divida a ningún otro término anterior. Por ejemplo, el factor 41 de F(20) no divide a ningún término anterior. En este caso le llamaremos factor característico o primitivo. Los tienes en https://oeis.org/A001578.

CONJETURA DE OPPERMANN

Esta conjetura está relacionada con otras tres que ya hemos estudiado, como son las de Legendre, Andrica y Brocard.

La primera afirma que *entre dos cuadrados consecutivos* n^2 *y* $(n+1)^2$ *existe siempre un número primo*, la de Andrica que "*La diferencia entre las raíces cuadradas de dos números primos consecutivos es siempre menor que 1*" y la de Brocard que "*Para n>1, si representamos como p(n) al enésimo número primo, se verificará que entre* $p(n)^2$ *y* $p(n+1)^2$ *existirán al menos cuatro números primos*".

En las entradas enlazadas se estudian las tres y sus relaciones mutuas.

Conjetura de Oppermann

Esta conjetura está muy relacionada con las tres referidas, y es una condición más fuerte que ellas. Fue establecida por Opperman en 1882.
Afirma lo siguiente:

Para todo número entero x>1, existe al menos un número primo entre $x(x-1)$ y x^2, y otro primo entre x^2 y $x(x+1)$.

El que tenga un carácter más fuerte proviene de que $\mathbf{x(x-1)>(x-1)^2}$ y $\mathbf{x(x+1)<(x+1)^2}$, con lo que los intervalos en los que se ha de encontrar un número primo se acortan.

Observamos que tanto x(x-1) como x(x+1) son números oblongos, y además consecutivos, siendo x^2 la media de ambos.
Al igual que nos ocurrió con la conjetura de Legendre, si usamos la función π, que da la distribución de los números primos (π(200) equivaldría a los primos que existen menores o iguales a 200), la conjetura de Opperman se podría expresar así:

$$\pi\big(n(n+1)\big) > \pi(n) > \pi(n(n-1))$$

Lo interesante aquí es que las desigualdades son estrictas, lo que indica que existen números primos intercalados, que es lo que afirma la conjetura.

Comprobación de la conjetura

Como en anteriores desarrollos de esta serie, usaremos nuestra herramienta conjeturas.xlsm, que puedes descargar desde la página
http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm

Esta hoja posee algunas funciones interesantes, aunque el trabajo de comprobación depende de nosotros. Hemos construido un esquema que nos permitirá la comprobación. Puedes intentarlo también. El nuestro es así:

Conjetura de Opperman

Valores de N	N(N-1)	Próximo primo	N^2	Próximo primo	N(N+1)
100	9900	9901	10000	10007	10100
101	10100	10103	10201	10211	10302
102	10302	10303	10404	10427	10506
103	10506	10513	10609	10613	10712
104	10712	10723	10816	10831	10920
105	10920	10937	11025	11027	11130
106	11130	11131	11236	11239	11342
107	11342	11351	11449	11467	11556
108	11556	11579	11664	11677	11772
109	11772	11777	11881	11887	11990
110	11990	12007	12100	12101	12210

En primer lugar se ha diseñado la cabecera, de forma que contenga los tres valores que figuran en la conjetura, N(N-1), N^2 y N(N+1). Entre ellos se han reservado dos columnas para que aparezcan los números primos que anuncia la conjetura.

La estructura es muy sencilla. Todo depende del número que escribamos en la parte superior izquierda, en la imagen el 100. Debajo de él figurarán automáticamente los siguientes. Esto no es necesario, bastaba con un número, pero así percibimos mejor la potencia de la conjetura. Hemos programado que cada celda sea igual a la anterior más una unidad.

Las columnas N(N-1), N^2 y N(N+1) son fáciles de rellenar en una hoja de cálculo y no las explicaremos. Las correspondientes a los primos que se esperan las hemos rellenado con la función PRIMPROX, que nos da el próximo primo mayor que un número. En la segunda columna aparecerá PRIMPROX(N(N-1)) y en la cuarta PRIMPROX(N^2).

Esta función nos da el primer primo entre esos números, pero con eso nos basta, ya que sólo deseamos resaltar que existe uno al menos. Si hubiera más, aparecería el primero de ellos.
Bastará ahora elegir números más pequeños o mayores para que verifiquemos la conjetura en casos particulares.

Conjetura de Opperman

Valores de N	N(N-1)	Próximo primo	N^2	Próximo primo	N(N+1)
2	2	3	4	5	6
3	6	7	9	11	12
4	12	13	16	17	20
5	20	23	25	29	30
6	30	31	36	37	42
7	42	43	49	53	56
8	56	59	64	67	72
9	72	73	81	83	90
10	90	97	100	101	110
11	110	113	121	127	132
12	132	137	144	149	156

Forzamos la hoja de cálculo con números mayores:

Conjetura de Opperman

Valores de N	N(N-1)	Próximo primo	N^2	Próximo primo	N(N+1)
30000	899970000	899970007	900000000	900000011	900030000
30001	900030000	900030011	900060001	900060013	900090002
30002	900090002	900090049	900120004	900120007	900150006
30003	900150006	900150029	900180009	900180059	900210012
30004	900210012	900210013	900240016	900240017	900270020
30005	900270020	900270043	900300025	900300043	900330030
30006	900330030	900330043	900360036	900360037	900390042
30007	900390042	900390047	900420049	900420061	900450056
30008	900450056	900450059	900480064	900480079	900510072
30009	900510072	900510089	900540081	900540083	900570090
30010	900570090	900570103	900600100	900600119	900630110

Si forzamos un poco más, ya no podemos contar con el cálculo en números enteros, y la hoja nos da error:

Conjetura de Opperman

Valores de N	N(N-1)	Próximo primo	N^2	Próximo primo	N(N+1)
50000	2499950000	#¡VALOR!	2500000000	#¡VALOR!	2500050000
50001	2500050000	#¡VALOR!	2500100001	#¡VALOR!	2500150002
50002	2500150002	#¡VALOR!	2500200004	#¡VALOR!	2500250006
50003	2500250006	#¡VALOR!	2500300009	#¡VALOR!	2500350012
50004	2500350012	#¡VALOR!	2500400016	#¡VALOR!	2500450020
50005	2500450020	#¡VALOR!	2500500025	#¡VALOR!	2500550030
50006	2500550030	#¡VALOR!	2500600036	#¡VALOR!	2500650042
50007	2500650042	#¡VALOR!	2500700049	#¡VALOR!	2500750056
50008	2500750056	#¡VALOR!	2500800064	#¡VALOR!	2500850072
50009	2500850072	#¡VALOR!	2500900081	#¡VALOR!	2500950090
50010	2500950090	#¡VALOR!	2501000100	#¡VALOR!	2501050110

Esto es normal y lo tenemos asumido. No pretendemos grandes cálculos, imposibles con el formato de coma flotante, sino crear esquemas que nos ayuden a entender mejor las conjeturas.

La conjetura afirma la existencia de un número primo, pero en la práctica pueden aparecer muchos más. En la imagen que sigue hemos usado la función PRIMENTRE, que también está incluida en la hoja Conjeturas, y se puede observar que el número de primos es considerable.

Valores de N	N(N-1)	Primos intercalados	N^2	Primos intercalados	N(N+1)
100	9900	9	10000	11	10100
101	10100	12	10201	11	10302
101	10100	12	10201	11	10302
102	10302	11	10404	11	10506
102	10302	11	10404	11	10506
103	10506	9	10609	12	10712
103	10506	9	10609	12	10712
104	10712	9	10816	12	10920
105	10920	9	11025	12	11130
106	11130	9	11236	13	11342
107	11342	11	11449	11	11556

Relación con la espiral de Ulam

Si observamos una imagen de la espiral de Ulam, nos daremos cuenta de que la conjetura que estudiamos viene a decir que cada lado de dicha espiral ha de contener un número primo.

101	100	99	98	97	96	95	94	93	92	91
102	65	64	63	62	61	60	59	58	57	90
103	66	37	36	35	34	33	32	31	56	89
104	67	38	17	16	15	14	13	30	55	88
105	68	39	18	5	4	3	12	29	54	87
106	69	40	19	6	1	2	11	28	53	86
107	70	41	20	7	8	9	10	27	52	85
108	71	42	21	22	23	24	25	26	51	84
109	72	43	44	45	46	47	48	49	50	83
110	73	74	75	76	77	78	79	80	81	82
111	112	113	114	115	116	117	118	119	120	121

Ya sabemos que pueden existir más. La imagen está tomada de nuestro documento

http://www.hojamat.es/sindecimales/divisibilidad/propuestas/rutas/htm/ulam.htm

que puede tener los vértices algo desplazados, pero se ven con claridad los distintos oblongos y cuadrados de cada lado y los primos comprendidos entre ellos.

Conjetura de Schinzel

Se puede afinar más la conjetura de Opperman. Schinzel conjeturó que *para x>8, existe al menos un número primo entre x y x+(lnx)²*.

Te invitamos a comprobarlo. En la imagen puedes ver cómo lo hemos hecho:

Valores de N	Primos comprendidos	$N+(\ln N)^2$
2000	7	2057,773718
2001	7	2058,781317
2002	7	2059,788913
2003	7	2060,796506
2004	6	2061,804095
2005	6	2062,811682
2006	7	2063,819264
2007	7	2064,826844
2008	7	2065,83442
2009	7	2066,841993
2010	7	2067,849563

CONJETURA DE COLLATZ

Ya se trató esta conjetura en el blog ***Números y hoja de cálculo***, pero desde el punto de vista de su experimentación en un Taller de Matemáticas

http://hojaynumeros.blogspot.com.es/2011/05/la-conjetura-de-collatz-en-un-taller-de.html

En esta ocasión se buscarán rutinas y funciones que nos ayuden a comprobar la conjetura para muchos números, así como encontrar sus cúspides y órbitas. Se ha escrito mucho sobre esta conjetura, por lo que aquí se desarrollará sólo ese aspecto.

Planteamiento

Para quienes no conozcan esta conjetura recordaremos su planteamiento:

Se toma un número entero positivo N cualquiera, por ejemplo el 13, y se le aplica la siguiente operación, a la que llamaremos función COLL(N):

- Si el número es par, se divide entre 2.
- Si el número es impar, se multiplica por 3 y se le suma 1.

En el caso del 13, como es impar, se le aplicará la segunda, y quedará COLL(13)=13*3+1=40.
La idea de la conjetura es que sigamos aplicando esta operación a todos los resultados que obtengamos, En nuestro caso sería COLL(40)=20 (por ser par), COLL(20)=10, COLL(10)=5, COLL(5)=3*5+1=16, COLL(16)=8, COLL(8)=4, COLL(4)=2, COLL(2)=1, y a partir del 1 se entra en el ciclo {4, 2, 1}

La conjetura afirma que este final en el 1 y el ciclo posterior **ocurre para cualquier otro entero positivo. Sea cual sea el comienzo, se llegará al número 1.** Todas las sucesiones construidas así terminarán en el ciclo 4, 2, 1.

Lo vemos con otro ejemplo:

COLL(6)=3, COLL(3)=10, COLL(10)=5, COLL(5)=16, COLL(16)=8, COLL(8)=4, COLL(4)=2, COLL(2)=1
Insistimos en que existen muchas publicaciones sobre esta conjetura y aquí sólo nos limitaremos a pequeñas comprobaciones. La más sencilla es mediante celdas en la hoja de cálculo.

Comprobación con celdas de hoja de cálculo

Escribimos el entero positivo inicial (lo podemos nombrar como *semilla*) en una celda, sea por ejemplo, la D4. En la celda inferior D5 escribimos SI(RESIDUO(D4;2)=0;D4/2;3*D4+1), que divide entre 2 el valor de la D4 si es par (porque entonces se verifica RESIDUO(D4;2)=0) y lo multiplica por 3 añadiendo 1 si es impar. En la imagen vemos el resultado para el número 132:

	C	D
1		
2		
3		
4		**132**
5		66

Al ser par el 132 se ha dividido entre 2. Ahora lo único que tenemos que hacer es rellenar esa fórmula hacia abajo y parar cuando aparezca un 1:

132	52
66	26
33	13
100	40
50	20
25	10
76	5
38	16
19	8
58	4
29	2
88	1
44	
22	
11	
34	
17	

(Troceamos la imagen porque aparecen muchos números)

La conjetura ha sido comprobada hasta números muy grandes, por lo que puedes tener la seguridad de que llegarás siempre al valor 1. Al conjunto de números que se recorren hasta llegar a ese valor le podemos llamar *órbita* del número dado, que aquí son los 29 números que aparecen en la imagen. Al número mayor que hayamos alcanzado en la órbita le llamaremos *cúspide*. En este ejemplo la cúspide es el mismo 132.

De esta forma tan simple podemos comprobar la conjetura dentro del alcance de la herramienta que usamos. Si la órbita tiene una longitud grande este procedimiento puede alargarse. Por ello acudiremos ahora a la definición de funciones:

Funciones sobre la conjetura de Collatz

Ya hemos presentado COLL(N). Sería bueno introducir su versión en VBA para poder construir sobre ella otras funciones más complicadas. Su código es muy sencillo:

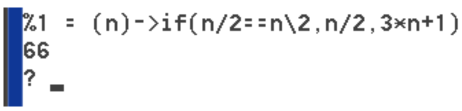
```
Public Function coll(n)
If n / 2 = n \ 2 Then coll = n / 2 Else coll = 3 * n + 1
End Function
```

No necesita grandes explicaciones. La condición n/2=n\2 equivale a indicar que n es par, ya que entonces el resultado de la división n/2 es idéntico al de la división entera n\2. El resto se entiende bien. Con esta función podemos reproducir las órbitas en columna de forma idéntica a como procedimos en el primer ejemplo.

En PARI el código es similar:

```
coll(n)=if(n/2==n\2,n/2,3*n+1)
```

En la imagen vemos el resultado de pedir coll(132)



Función orbicoll

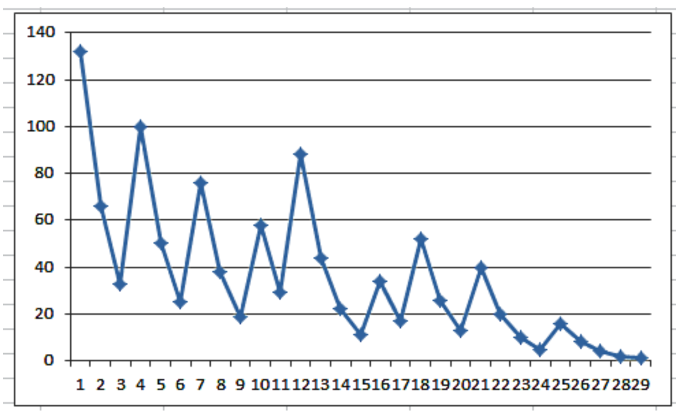
A cualquier número entero le podemos asignar una cadena que contenga todos los números por los que “pasa” hasta llegar al 1, es decir, su órbita. En VBA podía ser esta:

```
Public Function orbicoll(n)
Dim b
Dim s$ ‘Cadena (string) para recoger los resultados
b = n: s$ = Str$(b) ‘La cadena comienza con el número inicial (semilla)
While b <> 1 ‘Se trabaja hasta llegar al 1
b = coll(b) ‘ En cada paso se aplica la función COLL
s$ = s$ + Str$(b) ‘Se incorpora el resultado al string
Wend
orbicoll = s$
End Function
```

En la imagen puedes comprobar la creación de la órbita del número 132:

132
132 66 33 100 50 25 76 38 19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

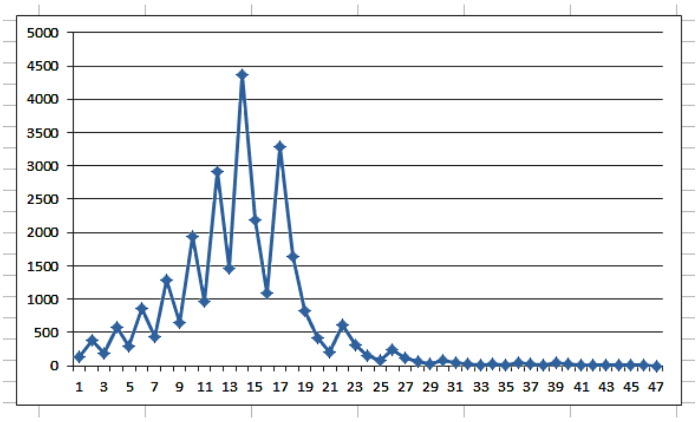
Después puedes usar la prestación de ***convertir texto en columnas*** (antes debes copiar la órbita en otra celda mediante *copiar valores*) y crear un gráfico que abarque toda la órbita:



Puedes observar que arranca en 132 y termina en 1. Los tramos ascendentes representan números impares, y los descendentes a los pares.

Puedes seguir este proceso con cualquier otro número entero y seguir la evolución de su órbita.

En la imagen aparece la órbita del número 127, más compleja y con una cúspide cercana a 4500:



Función orbicoll en PARI

Es fácil la traducción de esta función a PARI:

```
coll(n)=if(n/2==n\2,n/2,3*n+1)
orbicoll(n)=my(b=n,s=Str(n));while(b<>1,b=coll(b);s=concat(concat(s," "),Str(b)));s
```

En la imagen se ha pedido la órbita de 127:

```
127 382 191 574 287 862 431 1294 647 1942 971 2914 1457 4372 2186 1093 3280 1640
 820 410 205 616 308 154 77 232 116 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5
16 8 4 2 1
?
```

Estudia al código siguiente para la función *Iorbicoll*, que devuelve el número de elementos de una órbita:

```
Public Function lorbicoll(n)
Dim a, b
a = 1: b = n
While b <> 1
b = coll(b)
a = a + 1
Wend
lorbicoll = a
End Function
```

Con ella podemos comprobar lo que ya sabemos, que 132 tiene una órbita de 29 elementos.

Con la versión PARI puedes abordar casos con números mayores:

```
coll(n)=if(n/2==n\2,n/2,3*n+1)
lorbicoll(n)=my(b=n,a=1);while(b<>1,b=coll(b);a+=1);a
```

Te proponemos comprobar que el número 871 es el que posee la órbita de más longitud entre los de tres cifras, y que contiene 179 elementos. De los de cuatro cifras el de órbita de más longitud es el número 6171, con 262 elementos.

Función cuspicoll

Del mismo modo que construimos la órbita de un número entero positivo, podemos encontrar su cúspide. El procedimiento será similar, pero, en lugar de añadir resultados a un string, tomaremos nota en cada paso del máximo valor que ha aparecido:

```
Public Function cuspicoll(n)
Dim a, b
a = n: b = n
While b <> 1
b = coll(b)
If b > a Then a = b
Wend
cuspicoll = a
End Function
```

La instrucción clave es ***If b > a Then a = b***, que convierte a en el nuevo máximo si aparece un elemento b mayor que los precedentes.

Con las funciones definidas podemos construir un esquema en el que se analice el comportamiento de la conjetura de Collatz para una semilla dada:

Número	81
Órbita	81 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
Longitud	23
Cúspide	244

Finales previsibles

Algunos tipos de números presentan una órbita bastante previsible. Por ejemplo: Potencias de 2: a partir de ellos se entra en una ruta descendente y previsible que finaliza en el 1.

Números tipo (2^n-1)/3: Desembocan en una potencia de 2, por lo que también inician una ruta directa, y esto ocurre una potencia sí y otra no, porque 2^n es del tipo 3k+1 o 3k+2 y al multiplicar por Si es 3k+1 es candidato a que (2^n-1)/3 sea entero, y si es del tipo 3k+2, la siguiente potencia será 6k+4, o sea del tipo 3k+1

Otros tienen recorrido corto, como el 6, el 10 o el 20.

Es normal que pensemos en que muchas órbitas pasarán por ellos, y existan pares de órbitas que pasan ambas por el mismo punto de entrada, más o menos primario.

Podemos intentar ver si dos números presentan alguna coincidencia en sus órbitas, porque entonces compartirán final. No es difícil programar una función que nos devuelva un punto de coincidencia en las órbitas de dos números. En primer lugar necesitamos una función que nos indique si el número **n** pertenece a la órbita del número **m**. Puede ser esta:

```
Public Function enlacoll(m, n) As Boolean
Dim e As Boolean
Dim p
If m = n Then
e = True
Else
e = False
p = m  ‘p recorrerá la órbita de m
While Not e And p <> 1
p = coll(p)
If p = n Then e = True  ‘Si p es igual a n, sí pertenece
Wend
End If
enlacoll = e
End Function
```

Con esta función enlacoll(m,n) podemos saber si n pertenece a la órbita de m. Se puede organizar un esquema de cálculo:

m	57
n	52
¿Pertenece?	VERDADERO

En la imagen se ha verificado que 52 pertenece a la órbita de 57.

Coincidencia en las órbitas

Con la anterior función ya estamos preparados para encontrar la primera coincidencia entre dos órbitas. Si no existe una coincidencia anterior, se nos devolverá un cero. El código de la función es:

Public Function coincicoll(m, n)
Dim c, q
q = m
c = 0
While q > 1 And c = 0 ‘q recorre la órbita de m
If enlacoll(n, q) Then c = q ‘si q pertenece a la órbita de n, hay coincidencia
q = coll(q)
Wend
coincicoll = c
End Function

Con ella también podemos construir otro esquema de cálculo. En la imagen se comprueba que 125 y 126 comparten una subórbita de 95 elementos, que comienzan en 364.

m	125
n	126
Coincidencia	364
Órbita en común	95

Con estas ideas puedes construir otras muchas funciones, o emprender otras búsquedas. Sólo se ha pretendido en esta entrada dar ideas para comprobaciones y experimentaciones sobre la conjetura, ya de por sí bastante estudiada en otros aspectos.

CONJETURA DE RASSIAS

Esta conjetura recibe el nombre de su autor, M. Th. Rassias, que la enunció siendo muy joven, mientras preparaba una Olimpiada Matemática. Se puede formular de varias formas, pero la que preferimos es la siguiente:

Para cada número primo $p>2$ existen dos primos p_1 y p_2, con $p_1<p_2$ tales que
$(p-1)p_1=p_2+1$

Es decir, que si el primer primo lo multiplicamos por **p-1**, conseguimos un número al que precede otro número primo. Por ejemplo:

Para el número 17, el par de primos puede ser 2 y 31, porque (17-1)*2=32=31+1. Para el primo 47 los primos pueden ser 3 y 137, porque (47-1)*3=138=137+1

La conjetura afirma que siempre se pueden encontrar esos dos primos para uno dado.

Obtención con hoja de cálculo

En teoría se podría comprobar esta conjetura mediante una tabla de doble entrada con los primeros números primos, pero sería un procedimiento costoso en espacio y tiempo. Es preferible acudir al VBASIC o lenguaje similar. En Excel puedes intentar una función que nos devuelva el más pequeño **q** de los dos primos, suficiente para comprobar la conjetura, ya que el otro lo podemos calcular mediante ***(p - 1) * q – 1***

Public Function rassias(p)
Dim a, q

If Not esprimo(p) Then rassias = 0: Exit Function ‘Si no es primo nos devuelve un cero
q = 2 ‘Posible valor del primo más pequeño
a = 0 ‘Si a=0 significa que aún no se han encontrado los primos
While a = 0
If esprimo((p - 1) * q - 1) Then ‘prueba para saber que se encontraron los primos
a = q
End If
q = primprox(q) ‘Se prueba con el siguiente primo
Wend
rassias = a ‘Se encontró el primo menor
End Function

Las funciones ESPRIMO y PRIMPROX las puedes copiar desde nuestra entrada http://hojaynumeros.blogspot.com.es/2012/04/proposito-de-ormiston.html

Con esta función y el cálculo posterior podemos construir una tabla en la que para cada número primo contenga los dos primos más pequeños que verifican la conjetura. Sería similar a esta:

Primo dado	P1	P2
3	2	3
5	2	7
7	2	11
11	2	19
13	2	23
17	2	31
19	3	53
23	2	43
29	3	83
31	2	59
37	2	71
41	2	79
43	2	83
47	3	137
53	2	103
59	3	173
61	3	179

Programa en PARI

Para quienes conozcan el lenguaje PARI, con este programa comprobamos la conjetura para los primos inferiores a 200:

p=2;while(p<200,p=nextprime(p+1);q=2;a=0;while(a==0,b=(p-1)*q-1;if(isprime(b),a=q);q=nextprime(q+1));print(p,", ",a,", ",b))

Resultado:

```
? \r ini.txt
3, 2, 3
5, 2, 7
7, 2, 11
11, 2, 19
13, 2, 23
17, 2, 31
19, 3, 53
23, 2, 43
29, 3, 83
31, 2, 59
37, 2, 71
41, 2, 79
43, 2, 83
47, 3, 137
53, 2, 103
59, 3, 173
61, 3, 179
67, 2, 131
71, 2, 139
73, 5, 359
79, 3, 233
83, 2, 163
89, 3, 263
97, 2, 191
```

Con los cambios oportunos se puede lograr la comprobación para otros conjuntos de primos.

Otros puntos de vista

En la tabla anterior destaca la frecuencia con la que aparecen los valores 2 y 3 para el primo más pequeño. Es una indicación de que la conjetura no es algo complicado, sino que se comprueba fácilmente para valores pequeños. Podemos plantear una búsqueda para saber cuándo aparecerán otros valores, si es que lo hacen. Aquí tienes los resultados para la primera aparición de otros primos:

Primo menor	Primera ocurrencia
5	73
7	307
11	109
13	349
17	491
19	139
23	179
29	467
31	8689
37	1693

Esta tabla sugiere que la conjetura también se cumple para todo p_1. El problema radica en que no hay tope en la búsqueda de **p** y de $\mathbf{p_2}$, por lo que de no cumplirse para algún valor, entraríamos en un bucle sin fin. No obstante, lo intentamos con esta función:

```
Public Function rassias2(p)
Dim q, b, a

If Not esprimo(p) Then rassias2 = 0: Exit Function
q = 2
a = 0
While a = 0
b = (q - 1) * p - 1
If esprimo(b) Then
a = b
End If
q = primprox(q)
Wend
rassias2 = a
End Function
```

Con ella vemos que a todo valor de $\mathbf{p_1}$ le corresponde otro de $\mathbf{p_2}$. No tienen que resultar los mismos valores anteriores, porque al cambiar el punto de vista se encuentran otros mínimos, pero lo importante es que existe siempre una solución. Aquí tienes un resultado:

P1	P2	P
2003	8011	5
2011	4021	3
2017	12101	7
2027	12161	7
2029	4057	3
2039	20389	11
2053	73907	37
2063	12377	7
2069	12413	7
2081	20809	11
2083	12497	7
2087	33391	17

Por ejemplo, para 2081 como p_1, el valor de p_2 es 20809, calculado mediante p=11, ya que 20809=2081*10-1, y 20809 es primo.

No resistimos la elaboración de una función para p_2:

```
Public Function rassias3(p)
Dim q, b, a

If Not esprimo(p) Then rassias3 = 0: Exit Function
q = 2
a = 0
While a = 0
b = (p + 1) / q + 1
If esprimo(b) Then
a = b
End If
q = primprox(q)
Wend
rassias3 = a
End Function
```

Con ella se puede construir una tabla que relaciones $\mathbf{p_2}$ con $\mathbf{p_1}$ y **p**:

p2	p	p1
2	2	3
3	3	2
5	3	3
7	5	2
11	7	2
13	3	7
17	7	3
19	11	2
23	13	2
29	11	3
31	17	2

Todas estas tablas se podrían prolongar hasta números mucho mayores, y siempre existe una solución de dos primos respecto al dado, luego se puede dar por comprobada la conjetura dentro de la herramienta que hemos usado.

Primos relacionados con uno fijo

Por último, nos podríamos plantear si para cada valor de **p** podemos encontrar infinitos pares $\mathbf{p_1}$ y $\mathbf{p_2}$ que cumplan la conjetura. Lo dejamos como ejercicio. En la tabla observamos pares de seis cifras que cumplen la conjetura para **p=11**:

	Para p=11
p1	p2
100043	1000429
100109	1001089
100271	1002709
100361	1003609
100391	1003909
100523	1005229
100823	1008229
100937	1009369

Como los primos de la primera columna se multiplican por 10, los de la segunda terminan todos en 9. Este ejercicio lo podemos repetir para cualquier valor, y dentro del rango que deseemos. Terminamos con los primos de siete cifras que corresponden a p=137:

	Para p=137
p1	p2
1000037	136005031
1000367	136049911
1000397	136053991
1000547	136074391
1000577	136078471
1000763	136103767
1000793	136107847
1000889	136120903
1000973	136132327
1001177	136160071

Finitud de un conjunto

En comparación con las conjeturas explicadas en apartados anteriores, las que siguen, propias del autor, son meras distracciones sin trascendencia matemática, pero servirán para animar a otros a buscar situaciones similares.

El conjunto de números k tales que los valores de TAU(k) y PHI(k) son catetos en una terna pitagórica es finito.

En concreto, sólo parecen cumplir esta condición los números 20, 36, 60, 100 y 300. Publicados en http://oeis.org/A308664, Giovanni Resta la dio por verdadera en virtud de acotaciones conocidas para ambas funciones.

No es difícil abordar esta conjetura. TAU(k) es el número de los divisores de k. PHI(k) cuenta los números menores que k y coprimos con él (sin divisores comunes), incluyendo el 1.

Ambas funciones son populares, y están implementadas en todos los lenguajes de programación de tipo matemático. En concreto, en PARI se designan como ***numdiv*** (TAU) y ***eulerphi*** (PHI). De esa forma, la comprobación de la conjetura se puede obtener a través de este código:

for(i = 1, 2000, a = eulerphi(i); b = numdiv(i); if(issquare(a^2 + b^2), print1(i, ", ")))

Evidentemente, según la potencia del equipo de los lectores, se puede sustituir el tope 2000 por otro muy superior, como 10^6, lo que ralentizará algo el proceso. Para darlo como conjetura, el autor llegó a 10^22.

En la imagen hemos usado el tope de 10000 en la página de PARI/GP https://pari.math.u-bordeaux.fr/gp.html

```
? for(i = 1, 10000, a = eulerphi(i); b = numdiv(i); if(issquare(a^2 + b^2), print1
(i, ", ")))
20, 36, 60, 100, 300,
```

```
for(i = 1,  10000, a = eulerphi(i); b = numdiv(i); if(issquare(a^2 + b^2),
print1(i, ", ")))
```

Si se tiene tiempo de sobra, se puede usar el símbo ***oo*** como infinito:

for(i = 1, oo, a = eulerphi(i); b = numdiv(i); if(issquare(a^2 + b^2), print1(i, ", ")))

Hay que dejar el equipo funcionando e ir al cabo de un tiempo para ver si aparece otra solución. No aparecerá, porque según las acotaciones de Giovanni Resta, se tiene:

Para un número **n** suficientemente grande, tau(n)<n^(1/4), y phi(n)> n^(3/4), luego **phi** es mayor que **tau** para esos números, pero en una terna pitagórica el cateto mayor (en este caso PHI(n)), posee la acotación y<x^2/2 con el otro cateto, con lo que tendremos:

PHI(n)>N^(3/4), y por otra parte, PHI(N)<n^(2/4)/2, lo que es contradictorio.

Con hoja de cálculo también es fácil buscar estos números. Podemos usar estas funciones de VBasic y calcular, para cada par, la posible hipotenusa, para ver si es entera. Los siguientes listados están tomados del blog del autor http://hojaynumeros.blogspot.com/

```
Public Function euler(n)
Dim f, a, e
Dim es As Boolean

'Calcula la indicatriz de Euler de un número

a = n ‘Copia el valor de n
f = 2 ‘Inicia el listado de primos
e = n ‘Inicia el valor de PHI
While f <= a ‘Recorre los primos posibles
es = False ‘Variable que indica si hemos llegado a un divisor primo o no
While a / f = a \ f ‘Si es un factor, se va eliminando del valor de n
a = a / f: es = True
Wend
If es Then e = e * (f - 1) / f ‘Si se ha encontrado un factor primo, se incorpora a PHI
If f = 2 Then f = 3 Else f = f + 2 ‘Busca el siguiente primo
Wend
euler = e
End Function


Public Function tau(n)
Dim f, a, e, exx


a = n ‘Copia el valor de n
f = 2 ‘Inicia el listado de primos
e = 1  ‘Inicia el valor de TAU
While f <= a ‘Recorre los primos posibles
exx = 0
While a / f = a \ f
a = a / f: exx = exx + 1 ‘Incrementa el exponente del factor primo encontrado
Wend
e = e * (1 + exx) ‘Construye TAU
If f = 2 Then f = 3 Else f = f + 2
Wend
tau = e
End Function
```

Con ellos se pueden formar columnas y añadir otra con la hipotenusa, comprobando fácilmente que es entera en los números 20, 36, 60, 100 y 300 y en ningún otro.

Primos complementarios respecto a 2^k

En el año 2011, aprovechando que era un año primo, me pregunté si cada número primo se corresponde con otro de forma que su suma sea una potencia de 2. Mi respuesta de inicio fue afirmativa, es decir:

Para cada número primo p existe otro primo q tal que p=2^k-q para cierto valor de k natural.

Mis intentos con esta conjetura están reflejados en mi publicación *hojanum3*, resumen de mi blog en ese curso. Se puede descargar desde http://www.hojamat.es/publicaciones/Hojanum3.pdf. En el apartado "*Historia de una conjetura*" se cuentan mis procedimientos y la respuesta recibida de que la conjetura es falsa.

En principio todo es sencillo. Puedo llamar *comple2(p)* al menor número primo que sumado con *p* genera una potencia entera de 2, y estas sencillas dos líneas en PARI encuentran el complementario:

```
comple2(n) = for(m=1, oo, p=2^m-n;if(isprime(p), return(p)))
print(comple2(223))
```

En esta función se recorren exponentes de 2 desde 1 hasta infinito "oo", y se detiene cuando es prima la diferencia entre una potencia de 2 y **n**.

Si vamos cambiando la segunda línea, podremos encontrar el complementario. Hay que advertir que, por sencillez, no se ha exigido que el número sobre el que actuemos sea primo. En el caso del ejemplo lo es, y resulta:

comple2(223)=
3705346855594118253554271520278013051304639509300498049262642688253220148477729

Esto ya se vio en el año 2011. La novedad en este documento es el uso de PARI, que es muy sintético.

Si ahora intentamos encontrar el complemento de 1871, ya podemos esperar, porque con un equipo doméstico no lo lograremos.

Según el comentario de Jens Kruse Andersen *"Then there is no prime q such that 509203+q = 2^n."*, tampoco es buena idea intentarlo con 509203.

Resumiendo: nuestra conjetura inicial era falsa.

Triangulares formados por concatenación

En nuestra sucesión https://oeis.org/A226789 descubrimos que sólo existen 6 números triangulares cuyas cifras se forman como concatenación de dos números consecutivos n+1 y n

A226789 Triangular numbers obtained as the concatenation of n+1 and n.
10, 21, 26519722651971, 33388573338856, 69954026995401, 80863378086336

Se han comprobado los cálculos con tres lenguajes de programación distintos, y no se ha encontrado otro ejemplo. Por tanto, la conjetura sigue abierta.

Cubos con cifras crecientes y decrecientes

Existen publicadas muchas sucesiones en las que se exige que las cifras de un número sean crecientes (suele ser en sentido amplio, permitiendo repeticiones) o bien decrecientes. Tuvimos la idea de aplicar esa exigencia a los cubos, con la sorpresa de que para cifras crecientes sólo había un número limitado de términos:

0, 1, 8, 27, 125

Empleando diversas técnicas y lenguajes, no ha sido posible encontrar más ejemplos de cubos con cifras crecientes, por lo que se puede enunciar la conjetura:

Sólo existen cinco cubos perfectos con sus cifras crecientes en base 10.

Con hoja de cálculo podemos usar esta función, que detecta si las cifras son crecientes:

```
Public Function cifras_crecientes(n) As Boolean
Dim j, l
Dim c As Boolean


l = numcifras(n)
c = True
If l > 1 Then
j = l
While j >= 2 And c
If cifra(n, j) > cifra(n, j - 1) Then c = False
j = j - 1
Wend
End If
```

cifras_crecientes = c
End Function

Con ella, y un criterio para saber si un número es cubo perfecto, podemos buscar soluciones.

Function escubo(n)
Dim a
a = Int(n ^ (1 / 3) + 10 ^ (-6))
If a * a * a = n Then escubo = True Else escubo = False

End Function

En PARI se pueden plantear dos puntos de vista: en el primero usar una función que detecte la propiedad directamente en un número cualquiera, y en el segundo se trata de ir construyendo los cubos uno a uno y esperar a que se cumpla la condición en uno de ellos.

Función directa

ok(n)=digits(n)==vecsort(digits(n))&&ispower(n,3)

Esta función, aplicada a un número natural, indica si es un cubo *(ispower(n,3)*) y si sus cifras crecen (*digits(n)==vecsort(digits(n)*)).

Es muy directa y sencilla, pero resulta bastante lenta, por tener que probar todos los números, sean cubos o no.

Construcción de cubos

Para encontrar cubos es preferible construirlos desde 0. Esto es lo que efectúa esta otra variante:

for(i=0,10^7,m=i^3;if(digits(m)==vecsort(digits(m)),print(m)))

Es tres veces más veloz que la función anterior, porque la variable **m** siempre será un cubo.

Hemos llegado de varias formas hasta 10^24, sin detectar otro cubo con cifras crecientes, por lo que podemos dar por cierte la conetura hasta que alguien descubra un contraejemplo.

Cubos con cifras decrecientes

Mediante un proceso similar se pueden encontrar los cubos perfectos que presentan cifras decrecientes (en sentido amplio). En PARI vale todo lo explicado, pero hay que sustituir *vecsort(digits(m)),* por *vecsort(digits(m),,4),* que ordena las cifras en sentido contrario.

En este caso sí aparecen infinitos cubos, pero la conjetura que se deduce de ellos es distinta:

Cubos: 0, 1, 8, 64, 1000, 8000, 64000, 1000000, 8000000, 64000000, 1000000000, 8000000000, 64000000000, 1000000000000, 8000000000000, 64000000000000, 1000000000000000, 8000000000000000, 64000000000000000, 1000000000000000000, 8000000000000000000, 64000000000000000000, 1000000000000000000000,…

Conjetura:

Todos los cubos con cifras decrecientes en base10 pertenecen a uno de estos tipos: 0, 10^k, 8*10^k, 64*10^k, con k entero mayor o igual a cero.

Como la anterior, la hemos verificado hasta 10^24, por lo que también queda abierta hasta que se encuentre un contraejemplo.

Números de Fibonacci con cifras decrecientes

Otro ejemplo interesante es el de los números de Fibonacci con cifras decrecientes, porque parece que dan lugar también a una sucesión finita:

1, 2, 3, 5, 8, 21, 55, 610, 987

Para hoja de cálculo podemos usar nuestra función ESFIBO, que determina si un número es de Fibonacci o no.

Public Function esfibo(n) As Boolean 'Devuelve verdadero si N es de Fibonacci
Dim f As Boolean
Dim a

f = False
a = 5 * n * n + 4
If escuad(a) Then f = True
a = 5 * n * n - 4
If escuad(a) Then f = True
esfibo = f
End Function

Se basa en popular criterio para saber si un número pertenece a la sucesión de Fibonacci. Lo puedes consultar en *Gaussianos*:

https://www.gaussianos.com/algunas-curiosidades-sobre-los-numeros-de-fibonacci/

Si aplicamos esta función ESFIBO con CIFRAS_DECRECIENTES llegaremos al resultado presentado.

En estos casos de sucesiones finitas hay que avanzar bastante para plantear una conjetura. Por eso debemos pasar a PARI, que llega más lejos en sus cálculos:

ok(n) = digits(n) == vecsort(digits(n),,4) && (issquare(5*n^2+4) || issquare(5*n^2-4))

for(i=2,10^6,if(ok(i),print1(i,", ")))

Hasta 10^6 devuelve los mismos resultados que la hoja de cálculo, salvo el 1, que se da por supuesto:

```
? ok(n) = dígitos(n) == vecsort(dígitos(n),,4) && (issquare(5*n^2+4) || issquare(5*
n^2-4))
for(i= 2,10^6,if(ok(i),print1(i,", ")))
2, 3, 5, 8, 21, 55, 610, 987,
```

Le hemos exigido algo más, que llegue a 10^8, y no ha aparecido ningún ejemplo más.

Podemos intentar generar los números de Fibonacci según su definición. Con este otro algoritmo hemos recorrido los 10^4 primeros con el mismo resultado:

m=1;n=1;for(i=1,10^4,p=m+n;m=n;n=p;if(digits(p) == vecsort(digits(p),,4),print1(p,", ")))

En la sucesión A273046, el colaborador Charles R Greathouse IV, llega a la misma cloncusión de que es probable que no existan más ejemplos.

Es interesante que en una misma cuestión hayamos presentado tres sucesiones finitas. Es una especie de probabilidad decreciente, de tal forma que, al crecer mucho los números, la misma tiende a cero, perdiendo la posibilidad de aparición de ejemplos nuevos. Como esto es solo una explicación no matemática, se quedarán en conjeturas.

Número par y su próximo primo

Esta conjetura surge de una pregunta:

¿Podrá ser cualquier impar diferencia entre un número par y su próximo primo si ambos suman otro primo?

En nuestra publicación *Inevitables primos* está contenido el estudio completo a partir de esta pregunta.

Ver http://www.hojamat.es/publicaciones/primos.pdf

Hemos implementado una función que a cada número impar (no analizamos en ella si lo es o no) le hace corresponder el mínimo número natural que sumado con él produce un primo en el que la suma de ambos también es primo

```
Public Function difconprim(n)
Dim i, d, dd, p
i = 2
d = 0
While d = 0 And i < 10 ^ 6
p = primprox(i)
If esprimo(p + i) And p - i = n Then d = i
i = i + 2
Wend
difconprim = d
End Function
```

Recorre los números pares (variable i) hasta un tope de 10^6 (para números mayores habría que aumentarlo) y estudia si el próximo primo p cumple que p+i es primo y la diferencia entre ambos es el número n dado. No está completo ni optimizado el código. Sólo pretendemos establecer una conjetura. Aquí tienes la tabla para los primeros números impares:

d	i
1	2
3	8
5	24
7	216
9	182
11	468
13	114
15	1136
17	1956
19	2484
21	1130
23	1338
25	3138
27	3272
29	1332
31	12858

Por ejemplo, para la diferencia 21 el primer número par que la produce es el 1130, en el que se dan estos datos:

		Es primo
Diferencia	21	
Número par	1130	
Siguiente primo	1151	VERDADERO
Suma prima	2281	VERDADERO

Si a 1130 le sumamos la diferencia 21 se convierte en el número primo 1151, cuya suma con el anterior 1130+1151=2281, también es un número primo.

Puedes construirte un esquema similar. El problema que se presenta es que las hojas de cálculo se ralentizan cuando el valor buscado tiene muchas cifras. Así, entre los números impares menores que 100 la solución mayor es la correspondiente al 97, que es nada menos que 3240996. Lo puedes ver en este esquema:

		Es primo
Diferencia	97	
Número par	3240996	
Siguiente primo	3241093	VERDADERO
Suma de primos	6482089	VERDADERO

Para paliar esta lentitud hemos realizado también la búsqueda con PARI

difconprim(n)={local(i=2,d=0,p=2);while(d==0&&i<10^7,p=nextprime(i);if(isprime(i+p)&&(p-i==n),d=i);i+=2);return(d)}

{k=1;while(k<100,write("final.txt",k," ",difconprim(k));k+=2)}

Si tienes preparado en la misma carpeta un archivo de nombre "final.txt", este código te crea en él un listado similar al que sigue (hemos recortado la parte de los números anteriores a 100)

81	404860
83	990054
85	404856
87	370286
89	1467990
91	1468296
93	370280
95	370278
97	3240996
99	838250

Parece que nos podemos atrever a expresar una conjetura:

Cualquier número impar es diferencia entre cierto número y su próximo primo, en el caso en el que la suma de ambos también sea prima.

Diferencias en interprimos cuadrados

Un interprimo es aquel número compuesto que es promedio entre dos primos consecutivos.
Están publicados en https://oeis.org/A024675.
Los hemos estudiado también en nuestro blog
https://hojaynumeros.blogspot.com/2016/01/los-interprimos-1.html

Entre ellos son interesantes los que además son cuadrados (ver https://oeis.org/A075190). En ellos, las diferencias con su primo anterior o el posterior no pueden ser cuadradas. Si el interprimo es n^2 y su diferencia con el anterior es k^2, entonces el primo sería $n^2-k^2=(n+k)(n-k)$, lo que sería contradictorio.
Al comenzar a buscar ejemplos nos hemos preguntado si se recorrerán todas las diferencias posibles, en este caso no

cuadradas. Vemos 2, 3, 5, 6, 7, 8, 12,… ¿Estarán todas? Al principio es así, como se observa en esta tabla, en la que a cada diferencia le corresponde el mínimo interprimo cuadrado que la presenta:

Diferencia	Cuadrado mínimo
2	9
3	64
4	
5	144
6	625
7	324
8	2601
9	
10	154449
11	260100
12	1681
13	898704
14	27225
15	114244
16	
17	278784
18	223729
19	4410000
20	25281
21	12888100
22	4730625
23	1512900
24	4774225
25	
26	8208225
27	6130576
28	1121481
29	12744900
30	34586161
31	2433600
32	45360225
33	9784384
34	1271279025
35	64064016
36	
37	69956496

Nuestra conjetura para este caso es:

Todo número no cuadrado es diferencia entre un interprimo cuadrado y su primo más próximo.

En la tabla se ve que se cumple al menos hasta el 37. Con el siguiente código PARI podemos estudiarla para cualquier otro número mayor:

difcuad(n)= { local(i=2,m,v=0,p,q);

```
while(v==0&&i<10^6,m=i*i; p=m-precprime(m-1);q=nextprime(m+1)-m;if(p==n&&q==n,v=m);i+=1)
;return(v) }
{x=difcuad(50);print(x);print(sqrt(x))}
```

Sustituye el 50 por otro número cualquiera, y si el resultado es 0, cambia 10^6 por una potencia mayor. Aunque PARI es rápido, puedes tener que esperar un poco. Si nuestra conjetura es cierta, al final obtendrás el cuadrado $3973159089=63033^2$.

Restos posibles al dividir un primo y su número de orden

El resto de la división entera entre un primo y su número de orden puede presentar muchos valores distintos.

¿Aparecerán todos los restos si recorremos los números primos y los dividimos entre sus números de orden? En http://oeis.org/A004648 tienes su enumeración ordenada:

0, 1, 2, 3, 1, 1, 3, 3, 5, 9, 9, 1, 2, 1, 2, 5, 8, 7, 10, 11, 10, 13, 14, 17, 22, 23…

Al recorrer los primeros 1000 primos echamos de menos algún resto, como el 18 o el 20 ¿acabarán apareciendo? Para averiguar esto fijamos un número grande, como el 10^6, y para cada valor de resto que elijamos, por ejemplo ese 18 que no aparece, recorremos todos los primos menores que el tope y les calculamos su resto respecto al número de orden. Si aparece el que queremos, ya lo hemos encontrado; si no, aumentamos el tope. Lo podemos construir en el Basic de las hojas de cálculo:

Public Function primoresto(n)
Dim a, i, p, r
a = 2: i = 1: r = -1: p = -2 Iniciamos la lista de primos y la variable r a -1
While p <> n And i <= 10 ^ 6 Bucle hasta la solución o hasta el tope
p = a - i * Int(a / i) Buscamos el resto entre el primo a y su orden i
If p = n Then r = a Si el resto coincide con el número propuesto, ya tenemos solución
i = i + 1 Si no, avanzamos en la lista de primos
a = primprox(a)
Wend
primoresto = r
End Function

Si la función devuelve el valor -1, es que no se ha encontrado solución y hay que subir el tope. Con esta función y con Excel, que es una hoja rápida, hemos encontrado estos valores:

Posible resto	Mínimo primo que produce este resto
1	3
2	5
3	7
4	379
5	23
6	401
7	61
8	59
9	29
10	67
11	71
12	467
13	79
14	83
15	179
16	431
17	89
18	176557
19	191
20	24419
21	491
22	97
23	101
24	499
25	1213
26	3169
27	3191
28	523
29	229
30	3187
31	223
32	3203
33	8609
34	3163
35	251
36	176509
37	257
38	24509
39	263
40	3253
41	269
42	547
43	3347
44	1304867
45	293
46	571

Llama la atención el mínimo primo que presenta resto 18. Efectivamente, 176557 es el primo número 16049 y el cociente entre ellos es 11 y el resto 18, como cabía esperar. Más impresionante es el correspondiente a 44, nada menos que 1304867. Para avanzar más hemos traducido el algoritmo a PARI

```
resprime(n)={local(a,i,r,p);a=2;i=1;r=-1;p=-
2;while(p<>n&&i<=10^6,p=a%i;if(p==n,r=a);i+=1;a=nextprime(a+
1));return(r)}
{for(i=1,50,print(resprime(i)))}
```

Con él, subiendo el tope a 10^8, hemos descubierto que el resto 110 no aparece hasta el primo 514279133

¿Existirá siempre un número primo que produzca un resto igual a un número que elijamos? No lo sabemos. Lo dejamos como conjetura:

Conjetura: Para cada número natural n>1 existe un número primo P(k) que produce un resto respecto a k igual a n.