



# Implementación de la conjetura de Goldbach en python

Ester Quiñonez,  
Joel Juárez

Banco de Guatemala  
Universidad Rafael Landívar

Enero, 2022

# Índice

Objetivos

Definición problema

Algoritmo

Librería Seaborn

Resultados

Anexos

# Objetivos

## General

Computar y graficar algunos resultados que permitan entender la conjetura de Goldbach.

## Específicos

1. Aplicar los conocimientos adquiridos sobre programación en python para la resolución de problemas.
2. Explorar las funciones básicas de la librería gráfica de python *Seaborn*.

# Conjetura de Goldbach

La conjetura de Goldbach tiene sus orígenes en el año 1742, fue el matemático ruso Christian Goldbach quien afirmó que:

*Todo número par mayor que 2 puede escribirse como suma de dos números primos.*

Actualmente se le conoce como conjetura fuerte de Goldbach, ya que existe una versión débil que afirma:

*Todo número entero mayor que 5 se puede escribir como suma de tres números primos.*

Dicha versión débil fue "demostrada" por el Harald Helfgott, en publicaciones realizadas en el 2012-2013. La conjetura fuerte de Goldbach es uno de los problemas abiertos más antiguos de la matemática.

Ejemplos:

- ▶  $10 = 3 + 7 = 5 + 5 = 2 + 3 + 5.$
- ▶  $20 = 3 + 17 = 7 + 13 = 2 + 7 + 11 = 2 + 5 + 13$

# Algoritmo

- ▶ Se le pide al usuario que ingrese un número entero mayor a 2.
- ▶ Se crea una función que guarde los números primos que hay hasta un número entero dado.
- ▶ Se crean una lista que guarde las combinaciones posibles de dos números primos que sumados generen el número ingresado por el usuario.
- ▶ Se crean un lista que guarde las combinaciones posibles de tres números primos que sumados generen el número ingresado por el usuario.
- ▶ A partir de las listas generadas en los pasos anteriores se genera una gráfica mediante la librería **Seaborn**.

# Seaborn

Es una librería que funciona basada en *matplotlib*, pero a diferencia de ésta provee una interfaz de mayor calidad gráfica para la visualización de datos estadísticos. Para importar la librería usamos el comando:

```
import seaborn as sns
```

Para realizar un *scatter plot* usamos la función **Implot()** y podemos ejecutar alguno de los siguientes comandos:

```
graf = sns.scatterplot(data = , x = " ", y = " ")
```

Algunas funciones que se pueden utilizar para modificar la visualización de los datos:

- ▶ La instrucción **`sns.set_style('')`** permite cambiar el tema de fondo de la gráfica, seaborn ya cuenta con temas específicos.
- ▶ La instrucción **`sns.set(rc = 'figure.figsize':( , ))`** permite controlar el tamaño de la gráfica.
- ▶ Para realizar un histograma **`sns.distplot()`**.
- ▶ Para realizar gráfica de barras **`sns.countplot()`**



# Resultados

Para el entero 150:

$$11 + 139 = 150$$

$$13 + 137 = 150$$

$$19 + 131 = 150$$

$$23 + 127 = 150$$

$$37 + 113 = 150$$

$$41 + 109 = 150$$

$$43 + 107 = 150$$

$$47 + 103 = 150$$

$$53 + 97 = 150$$

$$61 + 89 = 150$$

$$67 + 83 = 150$$

$$71 + 79 = 150$$

$$2 + 11 + 137 = 150$$

$$2 + 17 + 131 = 150$$

$$2 + 41 + 107 = 150$$

$$2 + 47 + 101 = 150$$

$$2 + 59 + 89 = 150$$

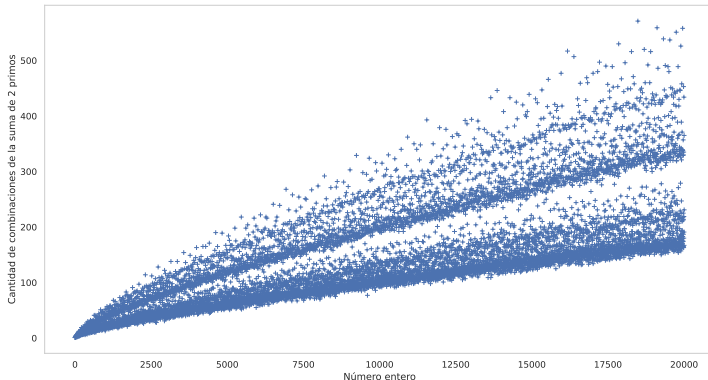
Se encontraron 12 combinaciones distintas de la suma de 2 primos y 5 para la suma de 3 primos.

Para el entero 79:

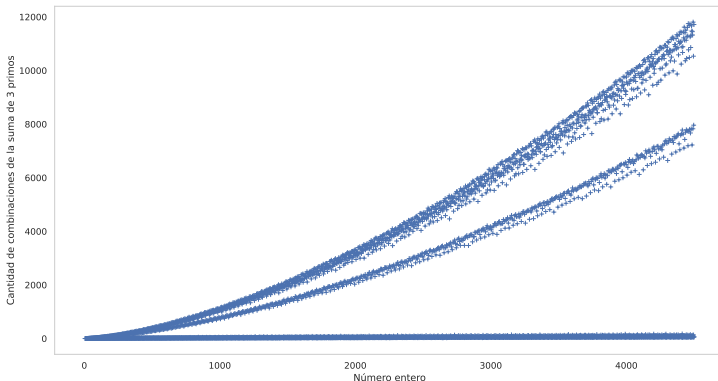
$3 + 3 + 73 = 79$	$5 + 37 + 37 = 79$	$13 + 19 + 47 = 79$
$3 + 5 + 71 = 79$	$7 + 11 + 61 = 79$	$13 + 23 + 43 = 79$
$3 + 17 + 59 = 79$	$7 + 13 + 59 = 79$	$13 + 29 + 37 = 79$
$3 + 23 + 53 = 79$	$7 + 19 + 53 = 79$	$17 + 19 + 43 = 79$
$3 + 29 + 47 = 79$	$7 + 29 + 43 = 79$	$17 + 31 + 31 = 79$
$5 + 7 + 67 = 79$	$7 + 31 + 41 = 79$	$19 + 19 + 41 = 79$
$5 + 13 + 61 = 79$	$11 + 31 + 37 = 79$	$19 + 23 + 37 = 79$
$5 + 31 + 43 = 79$	$13 + 13 + 53 = 79$	$19 + 29 + 31 = 79$

Al ser el 79 un número impar encontramos combinaciones únicamente de 3 números primos, con un total de 24 combinaciones distintas.

# Gráfica para la cantidad de combinaciones de la suma de dos números primos



# Gráfica para la cantidad de combinaciones de la suma de tres números primos



# Anexos

- ▶ Repositorio git:  
<https://github.com/PES-BG/proyecto-y-joel-ester>
- ▶ Acerca de Seaborn: <https://seaborn.pydata.org/index.html>