

2013

Primer Proyecto

Programado

Taller de Programación

Prof. Ing. Ericka Solano Fernández



Estudiantes:

- Pedro Rodríguez de Oliveira
- Kevin Alvarado Lamas
- David Obando Paniagua



Índice

Especificaciones del Proyecto	2
Diseño y Manipulación de Interfaz Gráfica	10
Investigación Hilos en Python	12
Cuadro de Alcances Logrados	13
Conclusiones Personales del Proyecto	14

Especificaciones del Proyecto

Entendiendo algunas curiosidades numéricas



Se desea generar un proyecto en el lenguaje de programación Python que permita mostrar algunas curiosidades que se descubren al trabajar con números y más que su resultado final, se desea presentar una guía de cómo fue producido este resultado.

Algunas de estas curiosidades se han trabajado en clase y otras serán asignadas para su trabajo individual.

La conjetura de Goldbach

En teoría de números, la conjetura de Goldbach es uno de los problemas abiertos más antiguos en matemáticas. A veces se le califica del problema más difícil en la historia de esta ciencia. Su enunciado es el siguiente:

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

Cabe notar que se puede emplear dos veces el mismo número primo.

$4 = 2 + 2$; $6 = 3 + 3$; $8 = 3 + 5$; $10 = 3 + 7$; $12 = 5 + 7$; $14 = 3 + 11 \dots$
Por ejemplo,

Números amigos

Se cuenta que en una ocasión le preguntaron a Pitágoras: “¿Para ti qué es un amigo?”. Y respondió: “Otro yo”. La aplicación de esta forma de concebir la amistad a la Teoría de los Números, condujo a decir que: dos números son amigos cuando cada uno de ellos es igual a la suma de los divisores del otro. Por ejemplo:

Divisores de 284: 1, 2, 4, 71, 142.

Divisores de 220: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110

$$284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110$$

$$220 = 1 + 2 + 4 + 71 + 142$$

Así, que 220 y 284 son números amigos.

El triángulo de Pascal

Es un árbol invertido o un triángulo donde todos sus bordes son 1 y los valores internos son la suma de los dos valores que se encuentren justo por encima de él.

$$\begin{array}{ccccccc}
 & & & & 1 & & \\
 & & & 1 & & 1 & \\
 & & 1 & & 2 & & 1 \\
 & 1 & & 3 & & 3 & & 1 \\
 1 & & 4 & & 6 & & 4 & & 1 \\
 & & & \vdots & & & & \ddots &
 \end{array}$$

La utilidad de construir un triángulo de Pascal se debe a su directa relación con el cálculo de los números combinatorios. A modo de ejemplo, calculemos los siguientes números combinatorios:

$$\binom{4}{0}, \binom{4}{1}, \binom{4}{2}, \binom{4}{3}, \binom{4}{4}$$

Estos números son iguales a 1 - 4 - 6 - 4 - 1 respectivamente. Es decir, son exactamente iguales a la quinta fila del triángulo de Pascal. En general, los números:

$$\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}$$

aparecen al escribir la $(n + 1)^{\text{ésima}}$ fila en el triángulo de Pascal.

Recuerde la fórmula de cálculo de combinaciones:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad n, k \text{ enteros positivos tales que } 0 \leq k \leq n$$

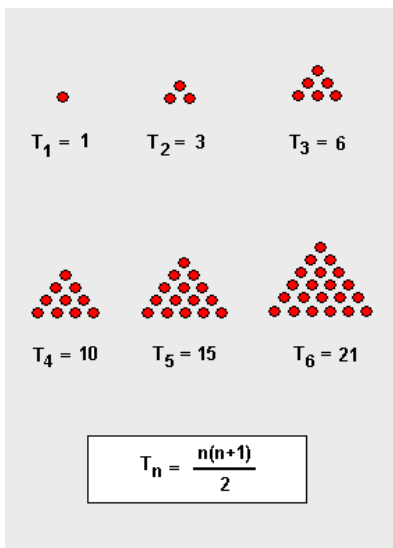
Y la fórmula de cálculo de la función factorial de N:

$$\begin{array}{ll}
 N! = 1 & \text{si } N=0 \\
 N! = N * (N-1)! & \text{Si } N > 0
 \end{array}$$

Número triangular

Es aquel que puede recomponerse en la forma de un triángulo equilátero (por convención, el primer número triangular es el 1).

Pitágoras y los Pitagóricos, consideraban sagrado el 10 escrito en forma triangular, y al que llamaban Tetraktys.



Curiosidades del 153

1.- Es el número más pequeño que puede ser expresado como la suma de los cubos de sus dígitos:

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$$

2.- La suma de sus dígitos es un cuadrado perfecto:

$$1 + 5 + 3 = 9 = 3^2$$

3.- Puede ser expresado como la suma de todos los números enteros del 1 al 17:

$$153 = 1 + 2 + 3 + 4 + \dots + 15 + 16 + 17$$

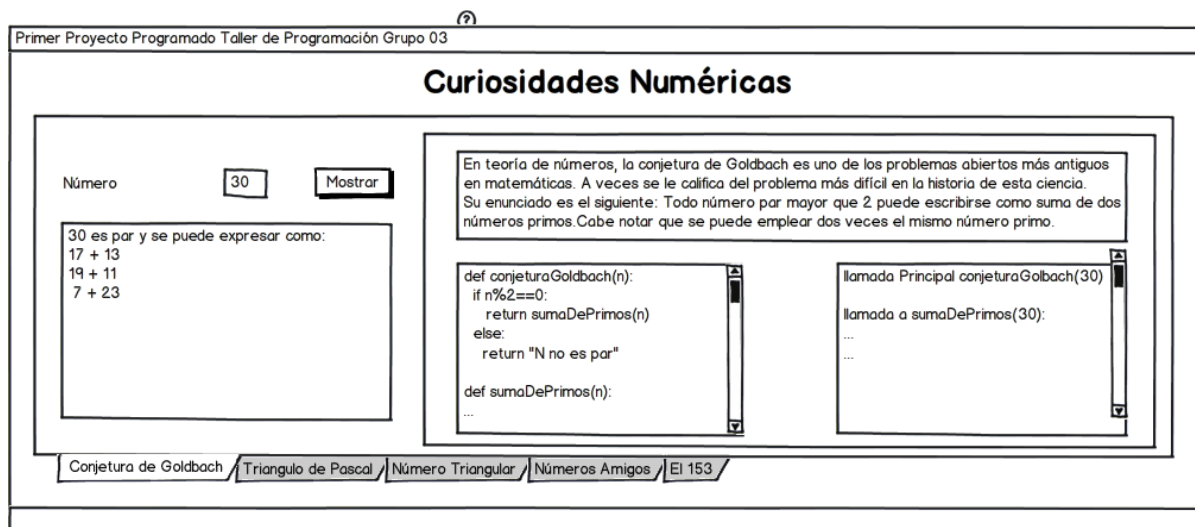
Esto significa que 153 es el decimoséptimo número triangular. Como su inverso, 351, también es un número triangular (suma del 1 hasta el 26) podemos decir que 153 es un número triangular invertible.

El proyecto

El proyecto se trata de construir una aplicación que permita mostrar el funcionamiento de estas situaciones por medio de casos de evaluación que se muestren en forma visual en la aplicación y que en un sector de la pantalla se presente el código de la implementación y la ejecución del código para el caso en cuestión.

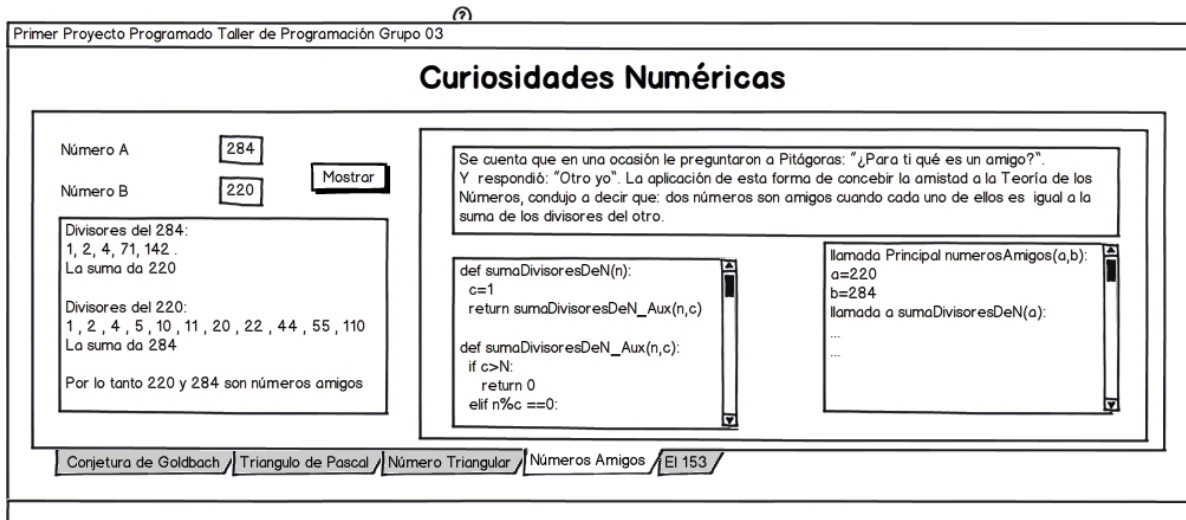
Esto quiere decir que por ejemplo para el caso de la conjetura de Goldbach el usuario puede suministrar un número X y la aplicación le mostrará en forma visual la forma en la que su código hallará la respuesta adecuada que es la suma de dos números primos que den como resultado el número par ingresado.

Un ejemplo de interfaz que muestre el funcionamiento de esta curiosidad es:

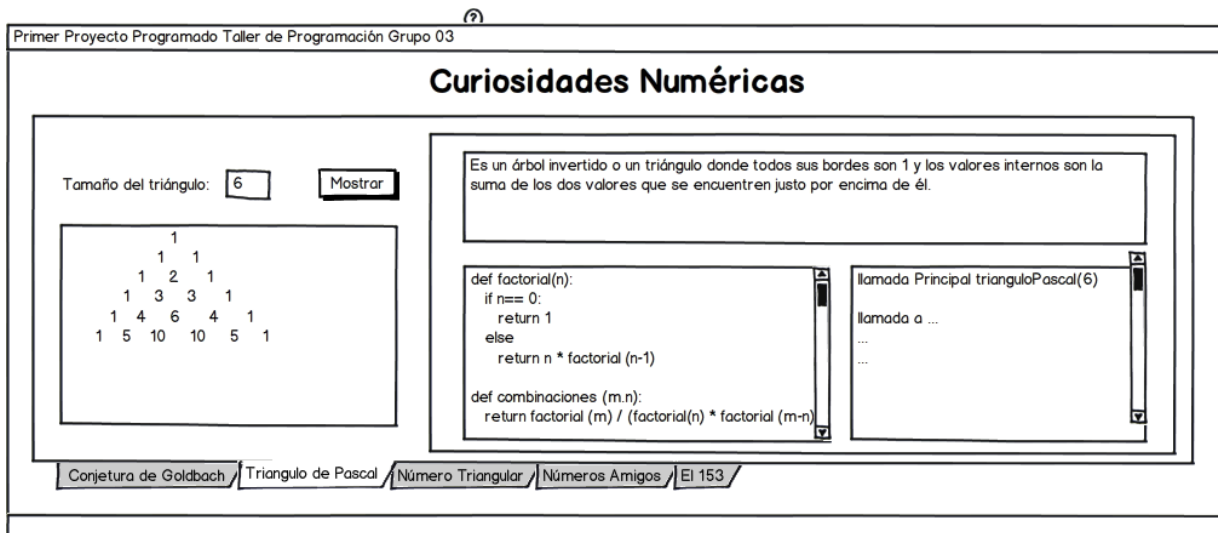


En el caso de los números amigos, el usuario proporcionará dos números enteros, la aplicación encontrará todos sus divisores y luego confirmará si la suma de los divisores del número A es igual a B y si la suma de los divisores de B es igual a A, mostrando en pantalla el paso a paso de cómo fue producida la respuesta.

Una posible apariencia para mostrar su funcionamiento podría ser:



En el caso del triángulo de Pascal el usuario suministra el número N que representa la cantidad de niveles que se desea generar en el triángulo y la aplicación muestra en este caso el triángulo generado de acuerdo a la especificación además de mostrar el paso a paso de cómo fue generado



En el caso del número triangular se presentarán dos opciones:

- Dado un número N averiguar si éste es triangular o no?
- Dado un número N mostrar el N-ésimo numero triangular, en este caso $2 \leq N \leq 20$.

En ambas situaciones, deberá al igual que las demás curiosidades mostrar el proceso realizado para alcanzar el resultado solicitado.

Un posible ejemplo de interfaz podría ser:

Primer Proyecto Programado Taller de Programación Grupo 03

Curiosidades Numéricas

Tamaño del triángulo:

El 6 es un número triangular pues con 6 se puede construir un triángulo equilátero.

```

      o
     oo
    ooo
   
```

Es aquel que puede recomponerse en la forma de un triángulo equilátero. Por convención, el primer número triangular es el 1. Pitágoras y los Pitagóricos, consideraban sagrado el 10 escrito en forma triangular, al que llamaban Tetraktys.

```

def esTriangular(n):
    ...

```

llamada Principal esTriangular(6)

llamada a ...

...

...

Conjetura de Goldbach / Triángulo de Pascal / Número Triangular / **Números Amigos** / El 153

Para la curiosidad del 153, se solicita que demuestre las tres características que se le han mencionado en este apartado, es decir mostrar porqué es un cuadrado perfecto, mostrar cómo es igual a la suma de los cubos de sus dígitos y porqué tanto él como su número invertido son números triangulares de modo que se pueda considerar un triángulo invertible.

Primer Proyecto Programado Taller de Programación Grupo 03

Curiosidades Numéricas

Curiosidades del 153

153 es igual a la suma de los cubos de sus dígitos:
 $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

La suma de sus dígitos es un cuadrado perfecto:
 $1 + 5 + 3 = 9$
 9 es un cuadrado perfecto porque $3^2 = 9$

La suma de los números del 1 al 17 = 153

1- Es el número más pequeño que puede ser expresado como la suma de los cubos de sus dígitos:
 2- La suma de sus dígitos es un cuadrado perfecto:
 3- Es triangular invertible:
 Esto es, es el 17avo número triangular y su inverso también es triangular: 351

```

def sumaCubosDigitos(n):
    suma=0
    return sumaCubos_aux(n,c)

def sumaCubos_aux(n,c):
    if n==0:
        return 0
    else:

```

llamada Principal sumaCubosDigitos(153)

suma=0

llamada a sumaCubos_aux(153,0)

...

...

Conjetura de Goldbach / Triángulo de Pascal / Número Triangular / **Números Amigos** / El 153

En todos los casos en la pantalla deberá aparecer siempre:

1. Una sección donde el usuario pueda conocer de qué se trata el problema,
2. Una sección que muestra cuál es el código en Python con el que fue resuelto, mostrando la/las funciones utilizadas para su solución.
3. Una sección donde se puede ver los resultados iniciales, intermedios y finales del proceso de ejecución de las funciones utilizadas.
4. Una sección donde el usuario puede ingresar los datos de prueba, iniciar la ejecución del proceso y ver el resultado final.

En el momento que se desee iniciar con la ejecución, se solicita que en el código de la función siempre se visualice la línea actual de ejecución, la cual puede ser resaltada de un color distinto al

resto de líneas para llevar la pista de cuál es la línea de código actualmente ejecutada. Para esto deberá proponer alguna forma de poder mostrar interactivamente cuando se ejecuta una condición, se invoca alguna función y por lo tanto se traslada el control a otro sector del programa tal y como ocurre realmente dentro del computador.

En la sección de resultados se mostrarán los distintos llamados a los métodos, los valores que toman los parámetros en cada llamada, los valores de las variables locales (si aplica) con el objetivo de por medio de la corrida comprobar y entender los distintos pasos por los que pasó el proceso para llegar a la respuesta final.

Consideraciones Administrativas

La especificación de este proyecto se libera el martes 10 de setiembre y tiene un tiempo de desarrollo establecido para tres semanas, por lo que su entrega formal será el viernes 4 de octubre del 2012 en hora de clase de taller para el grupo 3.

El proyecto se desarrolla en tríos y será entregado en formato de CD rotulado con el nombre de los participantes y la leyenda Curiosidades Numéricas, I Proyecto Programado, grupo 3.

La defensa del proyecto se realizará en tratos de 30 minutos por grupo por lo que las sesiones del 4 de octubre serán destinadas a la revisión. Cada grupo deberá presentarse a la hora que se le indique con el proyecto para su defensa.

El diseño y manipulación de la interfaz gráfica queda a total criterio de los equipos de trabajo.

Documentación Escrita

La documentación técnica deberá ser aportada en forma de comentarios dentro del código producido.

Cada una de las funciones implementadas deberá contar con el formato de documentación:

Nombre de la función, una breve descripción acerca del funcionamiento de la misma, detalle de entradas, detalle del tipo de salida, restricciones si las hay.

En cuanto a documentación escrita se debe aportar:

1. Esta especificación
2. Un apartado (no mayor de dos páginas) referente a las decisiones de diseño y manipulación de componentes de interfaz gráfica. Se constituye en un breve resumen de los componentes utilizados: canvas, entry, label, button, imágenes y cualquier otro elemento utilizado para lograr un diseño de interfaz gráfica agradable, y que le haya demandado investigación y pruebas, por ejemplo cómo emitir un sonido, cómo detectar cuándo se da click en alguna parte de la pantalla, etc).
3. Considere investigar sobre hilos para lograr el efecto que varios sectores de la pantalla puedan movilizarse al mismo tiempo.

4. Cuadro resumen de alcances logrados: esto es elaborar un inventario de todas las acciones que se desean lograr en este proyecto. Para cada uno de estos ítemes indique el porcentaje de logro (0-100%) y si es menor al 100% aporte una breve explicación del porqué no se alcanzó el ítem en su totalidad.
5. Conclusión Final: A título personal, cada uno de los participantes del equipo de trabajo deberá incluir un par de párrafos donde manifieste el grado de aporte que le otorgó la realización de este trabajo.

Documentación de Usuario

El equipo de trabajo construirá un video demostrativo del funcionamiento del proyecto, el cual no deberá sobrepasar los 2 minutos y sirva como una guía que le permita a un usuario inexperto comprender el funcionamiento del mismo y pueda ser útil para iniciar el uso apropiado de la aplicación.

Diseño y manipulación de componentes de Interfaz gráfica

La parte de la interfaz gráfica en el proyecto del Debugger la desarrollamos con la biblioteca de Python de Tkinter. En el proyecto utilizamos de esta biblioteca los siguientes elementos o widgets:

- **Ventana Inicial:** Ventana que aparece inmediatamente al ejecutarse el código, incluye el título y ciertos datos del proyecto. Es momentánea y solo dura unos segundos.
- **Ventana Principal:** Esta es sobre la cual está desarrollado el programa, la `top()`. Sobre ella se adaptaron los elementos como botones, cajas y entradas.
- **Menú:** Utilizamos el widget de Menú para colocar en la parte superior de la pantalla la opción de Salir y la de visualizar los créditos.
- **MenuButton:** El `MenuButton` como una herramienta distinta del Menú normal, lo preferimos para crear sobre sus diferentes botones las ventanas secundarias donde se encuentran cada una de las funciones.
- **ListBox:** Los `ListBox` fueron quizás los elementos más importantes del proyecto ya que en ellos desarrollamos propiamente el Debugger para ir ejecutando paso a paso las funciones, además también los utilizamos para presentar las variables y los resultados; esto por la facilidad que ofrecía de administrarse a modo de lista.
- **TextBox:** El `TextBox` también fue utilizado en el proyecto a modo de label, para colocar en él las instrucciones de cómo funciona cada función.
- **Scrollbar:** Utilizamos el `Scrollbar` como una necesidad obvia para las `List` y `Text Boxes`, para así ampliar el espacio sin necesidad de gastar más campo en la pantalla.

- **Button:** El Botón o Button lo utilizamos para ejecutar los comandos de las funciones y así recuperar de las Entries la información brindada por el usuario.
- **Label:** Aunque no las utilizamos en gran medida, fue un widget muy útil y sencillo de utilizar para colocar en las pantallas ciertas instrucciones cortas.
- **Entry:** El Entry fue vital ya que en ellos recuperamos los datos ingresados por el usuario a modo de string, pero el código aplicaba sobre ella un Int para que estuviese lista y ser utilizada por las funciones.
- **Image:** Solo utilizamos las imágenes para cargar la imagen de la ventana de inicio, y la imagen debe de ser una Gif y para mayor facilidad la colocamos sobre alguno de los otros widgets (buttons, canvas, etc).
- **TkMessageBox:** Fue uno de los widgets más prácticos para dar los resultados sin la necesidad de esmerarse en darlos en una ventana nueva, lo cual implicaría más inconvenientes.
- **Atributos:** Los atributos aunque fueron en su mayoría por estética, los utilizamos a lo largo de todo el proyecto para dar dimensiones, colores, entre otras características a las widgets.
- **Métodos:** El método mayormente usado fue el grid() para ubicar todos los objetos sobre la ventana principal, sin embargo también utilizamos place() con el mismo objetivo, y el pack().

Investigación Sobre Hilos en Python

No tuvimos una investigación ni un uso a fondo sobre el tema de Hilos, sin embargo los mismos fueron muy útiles para la implementación del Debugger . Pudimos mostrar en pantalla el código ejecutándose paso por paso y resaltando con un color distinto, las líneas de código que se ejecutaban en el momento; y al mismo tiempo mostrar en otra ListBox las variables que se iban utilizando o cambiando en el código y al final mostrar los resultados en otra ListBox.

Para utilizarlos requerimos de la importación de la biblioteca de Python llamada "threading" y de otra llamada "time" para pausar los pasos de la ejecución y que se notara el denotamiento de las líneas de código.

Cuadro de Alcances Logrados

Especificaciones del Proyecto	Porcentaje Completado
1. Diseño de funciones que permiten demostrar las 5 curiosidades matemáticas.	100%
2. Implementar un diseño de interfaz gráfica que permita mostrar los resultados de las funciones con datos ingresados por el usuario.	100%
3. Implementar un tipo de Debugger que muestre el código de la función en la pantalla y vaya ejecutando paso a paso la función, resaltando las líneas de código con un color distinto al fondo.	100%
4. Mostrar en la pantalla un durante el proceso de las funciones las variables que se utilizan en el código y mostrar al final los resultados.	100%
5. Mostrar en la pantalla un texto explicativo sobre lo que trata la función a ejecutar.	100%
6. Documentación o comentarios sobre cada una de las funciones utilizadas en el código del proyecto.	100%
7. Documentación Escrita del proyecto, el presente documento donde explicamos textualmente especificaciones del proyecto.	100%
8. Video Demostrativo de 2 minutos para orientar a un usuario inexperto en el uso del programa.	100%

Conclusiones Personales del Proyecto

El proyecto me pareció interesante en cuanto a lo que tenía que ver con la implementación del Debugger. Para mí más allá de desarrollar las funciones y los elementos de la interfaz en general, el mayor reto lo presento el efecto visual que había que representar para resaltar las líneas de código en las funciones.

Afortunadamente logramos administrar de buena manera el tiempo y el trabajo y finalizamos satisfactoriamente el proyecto. Por mi parte considero que personalmente tuve una participación no mayoritaria pero si sustancial. Fue un gran grupo de trabajo y logramos llevar a cabo el proyecto.

David Obando Paniagua

Mi experiencia con este proyecto programado fue muy gratificante, porque este fue mi primer proyecto con el cual trabajé enteramente con la librería de Tkinter. Al principio hubo que revisar muchos tutoriales para poder entender el uso de todos los elementos gráficos que compone a dicha librería y ciertos componentes de la librería TTK. Siento que mi aporte más fuerte fue en la creación de la interfaz gráfica, porque poco a poco se hicieron los componentes de las ventanas principales, llegando al resultado que se está mostrando.

Con la creación del proyecto, he podido sacar tres conclusiones principales. Primero, Tkinter es una librería muy completa para interfaces, pero sencilla en su uso. Por ejemplo, en Pygame, es necesario usar coordenadas para localizar algo en la ventana, mientras que Tkinter organiza los componentes de la ventana con métodos como grid y pack (hasta es posible usar el método de locación de Tkinter por medio del place). Además, Tkinter viene incorporado en el intérprete de Python. Después, el uso de la librería time es muy eficaz cuando se utiliza en programas orientados a interacción con el usuario, porque este necesita tiempo para analizar los resultados que imprime dicho programa. Por último, el uso de iteraciones fue muy útil en caso de querer demostrar una recursividad en el debugger. Este, como actúa de manera similar con la recursión, no ocupa ser expresado como una función auxiliar o extra, sino que puede ser un agregado a una función principal.

Pedro Rodríguez de Oliveira

El aporte dado a la tarea programa ha sido variada, ya que pude resolver 2 de los problemas planteados a su totalidad, y se investigó en varias páginas de internet los métodos para poder agregar funciones de Python y TKinter para poder realizar la parte de interfaz, como las ventanas, pestanas, botones, entre otros, necesarios para la tarea programada. Además logré construir el debugger con todo e interfaz gráfica para poder resolver las 3 curiosidades del 153 paso a paso hasta mostrar el resultado en pantalla. Como extra también implementé un método para agregar el efecto del Programa cargando, entre otras funciones.

Kevin Alvarado Lamas