Programación I - Primer Semestre 2018

Autor: Ing. Gonzalo de Soto - Docente de Prog. I - Sede Salto

# Consideraciones sobre la utilización de una alternativa de Interface Gráfica Básica para el Obligatorio

# Antecedentes:

El texto presente surgió a partir del pedido recibido de los estudiantes de Programación I de la Sede Salto sobre la posibilidad de utilizar una interfaz gráfica para interactuar y mostrar los resultados del primer trabajo obligatorio del curso.

Posteriormente, me transmitieron la existencia de planteos similares de estudiantes de la Sede Montevideo, lo que me motivó a hacerlo un poco más extenso y detallado.

Siendo importante dejar en claro que; lo que aquí se describe; constituye simplemente una referencia con algunas aclaraciones y ejemplos sobre las facilidades y forma de utilización de algunas de las alternativas que proporciona el paquete "graphics" escrito por John Zelle" para ser utilizado con el libro "Python Programming: An Introduction to Computer Science".

Y el objetivo es brindarles elementos que los ayuden a utilizar esas facilidades como herramientas para construir las primeras y sencillas componentes de una interfaz gráfica para el primer trabajo de programación obligatorio que van a desarrollar en su carrera, al tiempo que les sirva de introducción para una de las componentes de la informática que ha tenido más rápido crecimiento en los últimos años.

# Consideraciones previas sobre las interfaces gráficas.

Las interfaces gráficas surgieron como respuesta a la necesidad de mejorar la interacción entre los programas y sus usuarios, a medida que se extendía la utilización de la informática a un número creciente de disciplinas cuyo personal no tenía formación en programación.

Las primitivas interfases, definidas y utilizadas en general por las mismas personas que desarrollaban los programas, estaban basadas en el ingreso de comandos formados por secuencias de caracteres, mas parecidas a jeroglíficos incomprensibles que a instrucciones, para quienes no conocían el significado de los mismos.

Para avanzar en este sentido, se comenzaron a buscar alternativas basadas en la utilización y el despliegue de elementos gráficos en las pantallas de las computadoras, que permitieran una asociación sencilla y sin ambiguedades de las operaciones informáticas a ser ejecutadas, utilizando formas o imágenes de objetos o accesorios de la vida diaria, a través de cuya selección podían ser activados por parte de quienes los debían utilizar para sus respectivas actividades, sin necesidad de requerir una capacitación previa en informática.

También, se fueron sustituyendo progresivamente las formas de referenciar los elementos o las operaciones a ser activadas, basadas en nombres compactos de pocos caracteres o con siglas mnemotécnicas solamente entendibles por quienes las habían definido; por nombres completos que describían; de manera fácilmente comprensible; las componentes con las que estaban relacionadas, lo que ha contribuído de manera decisiva a facilitar su utilización a todos los niveles.

Las complementarias incorporaciones de imágenes, movimientos, sonidos, videos, etc., en numerosas interfaces gráficas, también hicieron su aporte para ampliar la capacidad, pero sobre todo la facilidad de comprensión y de interacción entre las aplicaciones y sus usuarios.

Y este proceso de creciente simplificación y clarificación en el uso de las aplicaciones informáticas; impulsado mediante las nuevas GUI; ha sido un factor importante de apalancamiento en el proceso de desarrollo y la creciente incorporación y utilización de estas tecnologías.

Al mismo tiempo, que se ha constituído en uno de los elementos que más ha contribuido a crear una espiral positiva para la difusión y adopción de este tipo de alternativas; en los sectores y las actividades más diversas; aún en aquellas que; hasta hace pocos años; era muy difícil de imaginar que pudieran decidirse su incorporación, debido principalmente a las dificultades en la comprensión que provocaban las anteriores interfaces.

Los nuevos enfoques de las GUI han permitido que, prácticamente cualquier persona; sea capaz de aprender a interactuar con las aplicaciones que le interesan, y que pueda utilizarlas de manera sencilla para la realización de sus tareas habituales.

# Las interaces gráficas en el aprendizaje de la programación

## - Una razonable contrapartida a los beneficios brindados por las GUI

La incorporación de interfaces sencillas; como instrumento para facilitar la utilización de diferentes tipos de instrumentos o tecnologías, tiene; como necesaria y lógica contrapartida; la necesidad de contar con herramientas cada vez más sofisticadas y complejas para poder desarrollarlas.

Y el caso de la incorporación de las nuevas interfaces gráficas en los desarrollos informáticos no es la excepción a lo antes mencionado, que han requerido el desarrrollo de numerosas y sofisticadas aplicaciones encargadas de manejar las complejas interrelaciones entre el sistema operativo y los programas, las cuales son necesarias para la implementación de estas interfaces.

Y como es imaginable, la utilización de estos programas exige una importante carga de capacitación adicional para poder aplicarlas adecuadamente.

Sin embargo, teniendo en cuenta las significativas ventajas proporcionadas por las interfaces gráficas en las aplicaciones informáticas actuales, resulta imprescindible para cualquier desarrollador contar con la formación necesaria para poder incorporarlas en sus desarrollos.

# Componentes básicos para el desarrollo de una interface gráfica

El desarrollo de interfaces gráficas requiere conocer:

1. Los diferentes tipos de accesorios gráficos (widgets) disponibles en la GUI a ser utilizada, a efectos de poder decidir cuáles de ellos serán los más adecuados para la interacción con los usuarios finales de cada una de las aplicacones.
2. La amplia gama de propiedades existentes para la configuración de cada uno de los accesorios, y elegir las alternativas requeridas en cada caso.
3. Las opciones disponibles para el gerenciamiento de los accesorios que van a integrar la interface diseñada, lo que permitirá configurar la interface de acuerdo al dieseño definido en cada caso.
4. Los conceptos a ser tenidos en cuenta para establecer una adecuada relación entre el usuario y la aplicación, en base a la cual se podrán definir los tipos de "eventos " que determinarán la "programación basada en eventos", con la que se desarrollan estas aplicaciones.  
   En esta concepción, el programador deberá definir todos los procedimientos que se activarán en el programa en función de los diferentes eventos que se pueden generar durante su ejecución.

De lo anterior se desprende que, a los efectos de contar con una adecuada formación en los diferentes aspectos vinculados a la utilización de interfaces gráficas, será necesario dedicar una importante carga horaria tanto para la comprensión de los conceptos teóricos, como para la realización de una gama de ejercicios que permita ponerlos en práctica.

Este requerimiento ha sido previsto en el plan de estudios, y la formación correspondiente se irá cumpliendo y consolidando en etapas sucesivas en las diferentes asignaturas de programación de las carreras de informática.

# El paquete "graphics", una interfaz gráfica para hacer las primeras experiencias

Para comenzar a conocer las características de las interfaces gráficas y poder realizar pruebas de forma gradual, se pueden utilizar las opciones que ofrece el paquete "***graphics***", mediante el cual se pueden implementar varias de las funcionalidades ofrecidas por la mayoría de las alternativas más complejas.

- Utilización de los accesorios gráficos disponibles.

Estas opciones incluyen objetos gráficos que permiten representar formas diferentes, mostrar texto, ingresar texto, mostrar imágenes, generar "movimiento" e implementar una interacción entre el usuario y el programa.

Y considerando,que para la mayoría de los estudiantes de Programación I, esta es su primera incursión en el terreno de las GUI, resulta de especial interés para todo el cuerpo docente de la materia que nos hagan conocer sus experiencias, comentarios y consideraciones con respecto a:

* las alternativas que eligieron para mejorar la presentación de los datos de un programa y lograr formas de interacción con elusuario
* el grado de dificultad y los principales problemas que encontraron para estas primeros implementaciones de una interfaz gráfica.

## Consideraciones sobre el paquete "graphics".

1. ***Autoría***: El paquete "graphics" fué escrito por John Zelle" para ser utilizado con el libro "Python Programming: An Introduction to Computer Science" (Franklin, Beedle &Associates).
2. ***Información sobre el paquete:*** Si bien en internet existen numerosas referencias y descripciones sobre este paquete, con diferentes grados de extensión y formas de explicar sus características, se sugiere el material que se encuentra en   
   " <http://mcsp.wartburg.edu/zelle/python/graphics/graphics/node2.html> " por tratarse de una versión resumida que incluye ejemplos que facilitan la comprensión.
3. ***Objetivos del paquete***: Tal como se describe en la "Introducción" incluida en el material mencionado en el punto anterior, este paquete es una biblioteca de objetos gráficos diseñada para facilitarle a los nuevos programadores la experiencia con gráficas en el computador, que es justamente el objetivo de este trabajo.
4. ***Instalación de "graphics"***:   
    Para la instalación de este paquete en una computadora con s.o. Windows,   
    seguir los pasos siguientes:
   * *Abrir la “****ventana de comandos de windows*** *==>* ***cmd*** *"*
   * *En la ventana de comandos digitar:* ***pip3 install –user*** [*http://bit.ly/csc161graphics*](http://bit.ly/csc161graphics) *y apretar la tecla* ***<Enter>***
   * *Se inicia el proceso de instalación que finalizará con un mensaje similar al siguiente:* ***Successfully installed graphics***

## Pasos para comenzar a conocer y aplicar las opciones disponibles en el paquete "graphics"

1. *Leer, "****toda****" la información descriptiva del paquete que se detalla en el sitio antes mencionado.*
2. *En el programa donde se va a utilizar se debe importar el módulo haciendo:* ***from graphics import \****
3. *Para cada uno de los elementos gráficos mencionados en el material:*
   1. *leer detalladamente las descripciones de sus respectivos objetivos, funcionalidades y características*
4. *probar el funcionamiento de cada uno de los elementos gráficos ingresando y ejecutando en el shell de Python los ejemplos que se detallan en el material*
5. *generar diferentes alternativas a los ejemplos incluídos en el material variando los valores y combinaciones posibles de los parámetros intervinientes y verificando los resultados obtenidos.*
6. *Anotar todas las dudas que puedan surgir al respecto para hacer las consultas correspondientes.*
7. *Abrir una ventana del editor de código Python para poder guardar en un archivo todo el código ejecutado en el Shell que les hayan resultado de interés, de manera de poder consultarlo y aplicarlo posteriormente en el trabajo obligatorio o en otros programas y obligatorios posteriores.*

## Componentes del paquete "graphics" que se sugieren como posibles alternativas para aplicar en la GUI del primer trabajo obligatorio

*Es importante aclarar previamente que queda a decisión de cada grupo de trabajo la decisión de incorporar o no una GUI como parte del programa a entregar para el primer trabajo obligatorio del curso.*

*Asimismo, en el caso de optar por incorporar una GUI, también queda a criterio de cada grupo la elección de la GUI que entiendan conveniente, siendo el paquete "graphics" una de las alternativas posibles a ser utilizadas.*

*El único requisito; en el caso de optar por incorporar una GUI; es que todos los integrantes del grupo sean capaces de demostrar, al momento de realizar la defensa del trabajo presentado, que conocen las características de la GUI elegida y que saben aplicar y definir las componentes de esa GUI que fueron utilizadas para el programa.*

*Las componentes del paquete "graphics" que se sugiere utilizar para una implementar una posible GUI en el programa son las que se indican en color "rojo":*

*a) ventana = GraphWin("Texto descriptivo", ancho, alto, autoflush=False)   
 Define la ventana de trabajo donde se agregan los componentes*

*b) ficha = Circle(Point(x, y), diametro)   
 Crea un círculo con centro en el punto de coord.(x,y) y con el diámetro indicado*

*c) columna = Rectangle(Point(x1, y1), Point(x2, y2)) #  
 Crea una rectángulo cuyo "Extremo Superior Izquierdo" es el punto de*

*coordenadas (x1, y1) y cuyo "Extremo Inferior Derecho" es el punto de*

*coordenadas (x2, y2)*

*d) texto = Text(Poin(x,y), "texto a mostrar")   
 Muestra en la ventana el texto detallado, centrado en el punto de coord.(x, y)*

*e) elemento.draw(ventana)   
 Método que "hace visible" en la ventana el elemento detallado.   
 Es importante tener claro que, para que los elementos definidos "aparezcan"   
 en la ventana luego de ser definidos, debe ejecutarse este método.  
 O sea, que no basta con definir un elemento para que el mismo aparezca,   
 debiendo ejecutarse este método para que eso ocurra.*

*f) elemento.move(dx, dy) #  
 Método que desplaza un elemento de acuerdo a los valores de desplazamiento   
 de sus coordenadas "x" y "y", según los valores indicados por "dx" y "dy".   
 Para lograr una visualización del "movimiento" se debe realizar un desplazamiento  
 progresivo utilizando alguna de las opciones de iteración como "for" o "while".   
 Por ejemplo, si un elemento debe desplazarse desde la posición (1,1) hasta   
 la posición (100, 100), se puede hacer una iteración tal que en cada una de las   
 iteraciones el elemento se mueva 5 lugares en cada coordenada (elemento.move(5,5)).  
  
 Y, en el caso de querer simularse trayectorias de diferente tipo, se pueden generar   
 diferentes variaciones de los desplazamientos para lograrlo.   
 Por ej., si quiero que se desplace primero horizontalmente y luego verticalmente,  
 se hace primero una iteración variando solamente la componente x   
 (elemento.move(dx, 0)) hasta llegar a la coord. x final, y luego se hace otra iteración  
 variando solamente la componente y (elemento.move(0, dy)) hasta llegar a la   
 coord. y final.*

*g)* ***Interacción entre el usuario y el programa:***

*La interacción entre el usuario y el programa se logra mediante el método   
 "ventana.getMouse()" que actúa de forma similar a la sentencia "****input()****" utilizada en  
 los programas Python para detener la ejecución del programa y habilitar al usuario a  
 "ingresar" datos.*

*En el caso del método "ventana.getMouse()", se detiene la ejecución del programa   
 hasta que el usuario haga "click" en alguna parte de la ventana, y el programa   
 recibe un objeto "Punto" del cual se pueden obtener sus "coordenadas" para determinar   
 en qué parte de la ventana hizo click el usuario.*

*Ejemplo: Si en el programa se codifica la sentencia "punto = ventana.getMouse()",   
 cuando el intérprete ejecuta esa sentencia, el programa se va a deterner a la  
 espera de que el usuario haga click en alguna parte de "ventana".   
 Luego que el usuario hace click, la variable "punto" queda asociada a un objeto  
 "Punto", del cual se pueden obtener sus coordenadas con los métodos siguientes  
 métodos "get"  
 coordX = punto.getX()  
 coordY = punto.getY()*

*Para probar este método se sugiere hacer un pequeño programa y ejecutarlo.*

*Ejemplo: from graphics import \**

*ventana = GraphWin("Prueba getMouse()", 600, 400, autoflush=False)*

*while True:*

*punto = ventana.getMouse()*

*print('Se hizo click en el punto de coord. x = ', punto.getX(), ' y = ', punto.getY())*

*Nota: Se utiliza "while True" para hacer tantas pruebas como se quiera. Para cancelar la   
 ejecución hacer "Ctrl+C".*

*h) Implementando un "botón"*

*Para los que ya conocen algunas interfaces gráficas les puede llamar la atención que no   
 haya un elemento "Botón" que permita definir alternativas para ejecutar diferentes  
 componentes del programa.*

*Sin embargo, de hecho existe y está directamente relacionado con el método anterior   
 "getMouse()", el cual puede ser considerado conceptualmente como un botón del   
 tamaño de un pixel.   
 Porque, ¿cual es el objetivo de un botón?. Habilitar la ejecución de cierto código cuando   
 el usuario hace click sobre la parte de la ventana donde está dibujado el botón.*

*En este caso, tendríamos un botón del tamaño de un pixel, lo cual hace bastante difícil para los usuarios el tener que "embocar" el cursor en el pixel apropiado.*

*Pero, por suerte, resulta muy fácil, utilizando el método "getMouse()" y algunos de los elementos de "graphics" que mencionamos anteriormente, poder "implementar un "Botón" del tamaño, con el texto y con los colores que nosotros queramos y asociarlo con la ejecución del código que nos interese.*

*Para eso procedemos de la siguiente forma:*

*a) Definimos y dibujamos un rectángulo en la parte de la ventana donde queremos ubicar nuestro "botón", recordando las coordenadas que le asignamos.*

*b) Dentro del rectángulo definimos y dibujamos el texto que queremos que tenga este botón.*

*c) Agregamos los colores que nos interesen para el fondo del rectángulo, su borde y el texto, asíi como el tamaño y el tipo de letra elegido.*

*d) ¿Cómo funciona este botón?  
De la misma forma que cualquier botón en otras GUI, podemos hacer que este botón implementado con los elementos de "graphics" cumpla la misma funcionalidad teniendo en cuenta lo siguiente:*

*- cuando se activa el método "getMouse()" mediante una sentencia   
 "punto = ventana.getMouse()"  
 el usuario puede hacer "click" en cualquier parte de la ventana*

*- Cuando el programa recibe el Punto(x,y) donde el usuario hizo "click", el programa deberá verificar si las coordenadas del punto donde se hizo click se encuentran "dentro" de las coordenadas del rectángulo que representa nuestro botón.   
Y, en el caso de que el click haya sido realizado dentro del rectángulo de nuesto botón, se ejecutarán las sentencias asociadas a ese botón.*

*Así de sencillo y mucho más acorde para un futuro informático, ya que de esta manera, no solamente utilizan un "boton"; como podría hacer cualquier usuario; sino que estarían "implementado" un botón.*

*En este punto damos por terminada esta breve introducción a la definición de una interfaz gráfica básica, para que aquellos grupos que quieran utilizarla puedan; utilizando el material indicado en internet, y las referencias y ejemplos antes mencionadas, elaborar su propia versión para este primer trabajo obligatorio o para la resolución de cualquier otro ejercicio u obligatorio posterior.*

## *Un último comentario respectoa las interfaces gráficas y los programas*

*Es fundamental que todos los estudiantes que están haciendo sus primeros pasos en una carrera en informática tengan muy claro que, lo más importante de cualquier programa no es su interface, sea del tipo que sea, sino lograr que el programa produzca el o los resultados esperados, a partir de la ejecución de los algoritmos y las estructuras de datos asociadas, que se definieron y codificaron para ello.*

*Y si bien las interfaces gráficas facilitan la interacción del usuario con las funcionalidades del programa, así como la visualización, presentación y comprensión de los resultados producidos, lo que las convierte en la parte más visible de los programas informáticos, las mismas cumplen una función complementaria de las partes del programa que procesan los datos y generan los resultados que son el objetivo final de todo programa.*