

Paradigma OOP – Tkinter - Pygame  
Trabajo Práctico N° 08

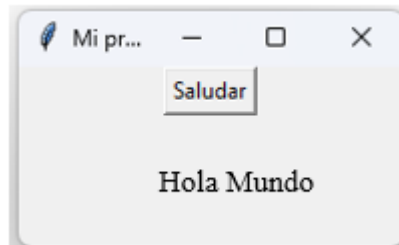
### Consideraciones Generales

- Crear el código fuente y guardarlo en un archivo del tipo apellido nombre tp08ej99.py  
(Los ceros representan la numeración que les ira dando acorde al número de Práctico y ejercicio.)
- La salida por pantalla debe contener toda la información necesaria para el usuario, a fin de que sea entendible el programa.

### Actividades

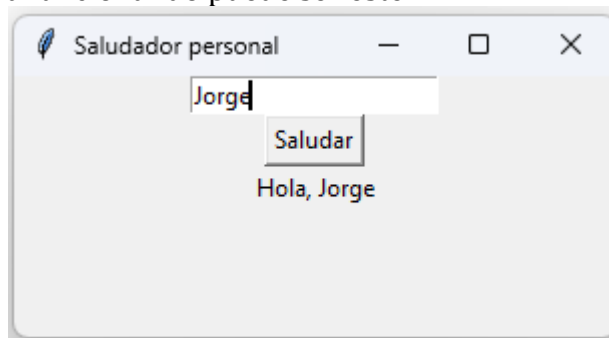
1. Hacer un programa que muestre una ventana con un botón “Saludar” y que al presionar dicho botón se imprima en una etiqueta la palabra “Hola Mundo”

Un ejemplo del programa funcionando puede ser este:



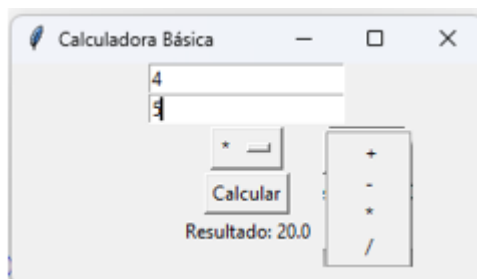
2. Hacer un programa que muestre una ventana con un botón “**Saludar**” y que al presionar dicho botón se imprima el contenido de la caja de texto en un etiqueta, por ejemplo si ingresa un nombre, mostrará el nombre

Un ejemplo del programa funcionando puede ser este:



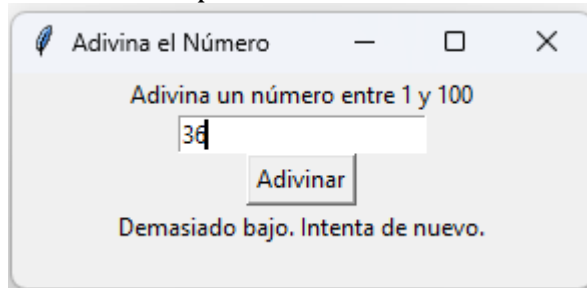
3. Hacer un programa que simule una **calculadora simple** con las operaciones de suma, resta, multiplicación y división. Deberá ingresar dos números, tener un lista de opciones para las operaciones y mostrar el resultado

Un ejemplo del programa funcionando puede ser este:



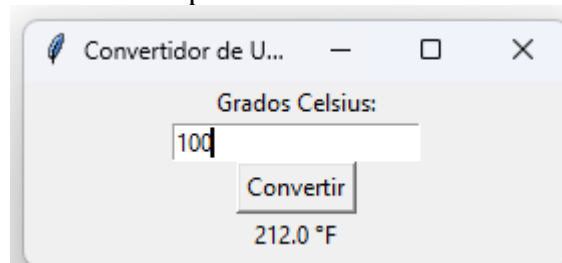
4. . Hacer un programa donde el usuario debe adivinar el numero que se generó aleatoriamente (Programa **Adivina Numero**). El programa deberá darle pistas al usuario si el numero ingresado es alto o bajo o si acertó.

Un ejemplo del programa funcionando puede ser este:

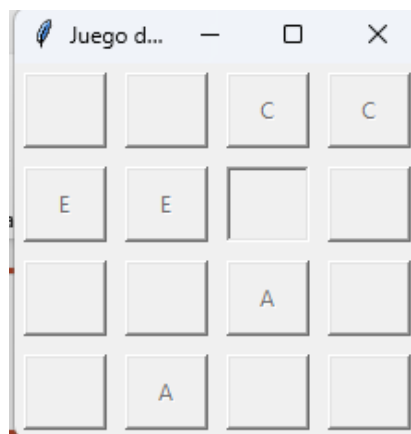


5 . Hacer un programa que simule una **calculadora de conversión** de unidades de Grados centígrados a Celsius. El programa podrá tener más unidades de conversión.

Un ejemplo del programa funcionando puede ser este:



6. Realizar un programa que simule al juego “buscaminas”, pero debe buscar la coincidencia de dos letras iguales “**buscaletras**”. El usuario deberá ir haciendo clic en los diferentes botones y el programa mostrará la letra que tiene oculta solo por un segundo y después desaparecerá. Cuando existan dos coincidencias de dos letras iguales el programa mostrará los botones con las letras encontradas en forma permanente. Un ejemplo del programa funcionando puede ser este:



7. Hacer un programa que simule una “**liquidación de sueldos**” para los empleados de una fábrica. La aplicación deberá permitir ingresar datos relevantes para calcular el sueldo bruto, los descuentos correspondientes y el sueldo neto final. La moneda a utilizar será en pesos argentinos (ARS).

Requerimientos del Programa:

- Ventana Principal: La aplicación debe mostrar una ventana principal con los siguientes campos y opciones:

- **Antigüedad en la Empresa (en años):** Un campo para ingresar la cantidad de años que el empleado ha trabajado en la fábrica.
- **Categoría del Empleado:** Un menú desplegable (Combobox) que permita seleccionar la categoría del empleado. Las categorías disponibles serán "A", "B" y "C". Cada categoría tendrá un valor base predefinido:
  - Categoría A: ARS 5000
  - Categoría B: ARS 7000
  - Categoría C: ARS 9000
- **Cantidad de Horas Trabajadas:** Un campo para ingresar la cantidad de horas normales trabajadas por el empleado durante el mes.
- **Valor por Hora Trabajada:** Un campo para ingresar el valor en pesos argentinos (ARS) de cada hora trabajada.
- **Cantidad de Horas Extras:** Un campo para ingresar la cantidad de horas extras trabajadas durante el mes.
- **Valor por Hora Extra Trabajada:** Un campo para ingresar el valor en pesos argentinos (ARS) de cada hora extra trabajada.
- **Otros Beneficios:** Un campo para ingresar cualquier otro monto adicional que el empleado reciba, como bonos, incentivos, etc.

**Cálculos Automáticos:**

- **Sueldo Base:** Debe calcularse como el valor base según la categoría, más un extra de ARS 200 por cada año de antigüedad.
- **Sueldo Normal:** Se calcula multiplicando las horas trabajadas por el valor de cada hora.
- **Sueldo Extra:** Se calcula multiplicando las horas extras por el valor de cada hora extra.
- **Sueldo Bruto:** Es la suma del sueldo base, sueldo normal, sueldo extra, y otros beneficios.

**Descuentos:**

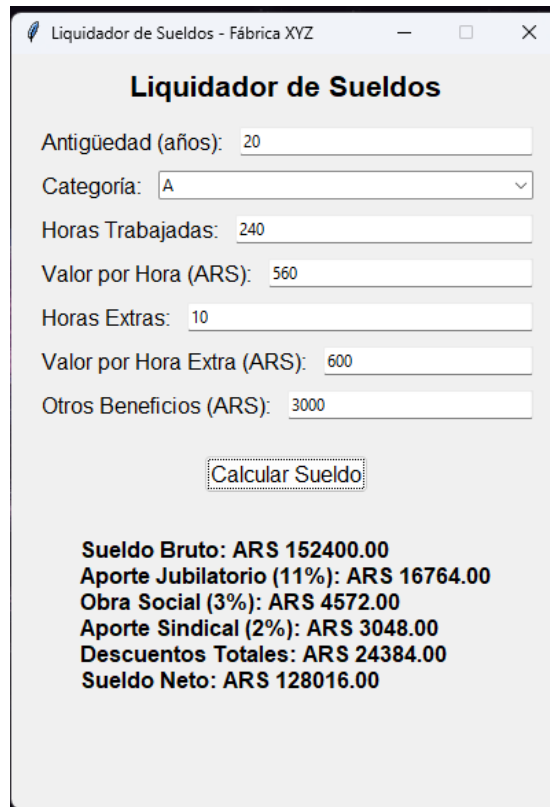
- **Aportes Jubilatorios:** Se descuenta un 11% del sueldo bruto.
- **Obra Social:** Se descuenta un 3% del sueldo bruto.
- **Aporte al Sindicato:** Se descuenta un 2% del sueldo bruto.
- **Sueldo Neto:** Se obtiene restando los descuentos totales del sueldo bruto.

**Botón de Calcular:** Un botón que permita realizar todos los cálculos mencionados y mostrar el resultado final en la misma ventana.

**Mostrar Resultados:** Los resultados deberán mostrarse en la parte inferior de la ventana e incluir:

- Sueldo Bruto
- Descuentos detallados (Aporte Jubilatorio, Obra Social, Aporte Sindical)
- Descuentos Totales
- Sueldo Neto

Un ejemplo del programa funcionando puede ser este:



**Liquidador de Sueldos**

Antigüedad (años):

Categoría:

Horas Trabajadas:

Valor por Hora (ARS):

Horas Extras:

Valor por Hora Extra (ARS):

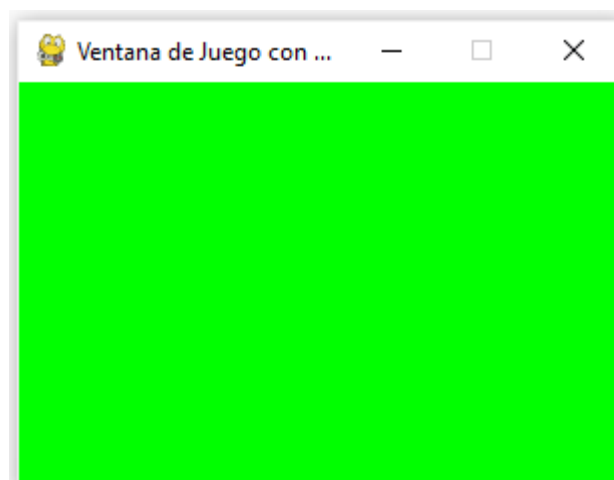
Otros Beneficios (ARS):

**Sueldo Bruto: ARS 152400.00**  
**Aporte Jubilatorio (11%): ARS 16764.00**  
**Obra Social (3%): ARS 4572.00**  
**Aporte Sindical (2%): ARS 3048.00**  
**Descuentos Totales: ARS 24384.00**  
**Sueldo Neto: ARS 128016.00**

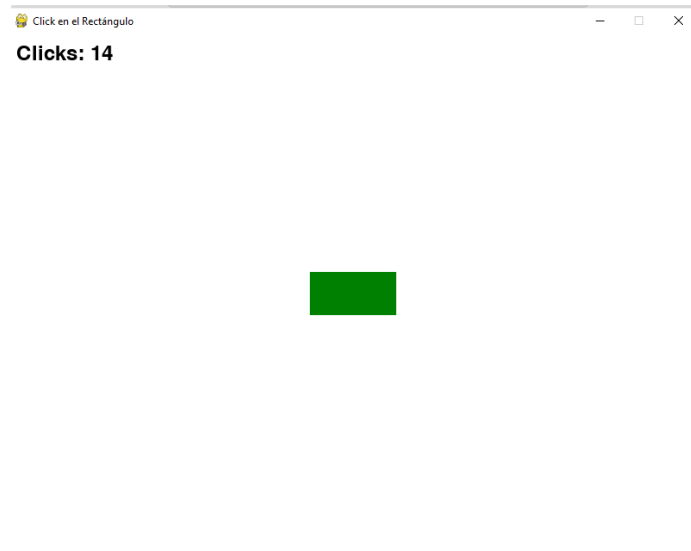
---

## Actividades con Pygame

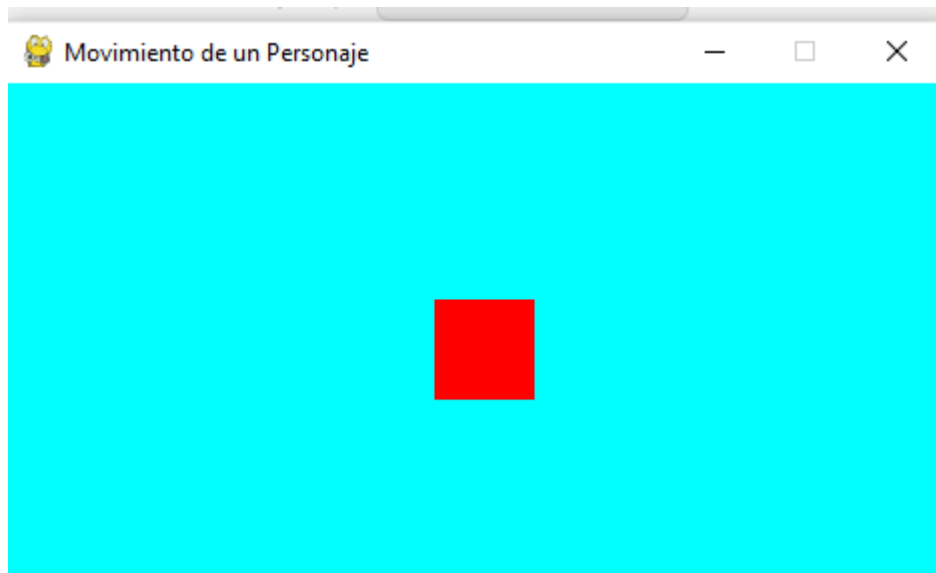
8. Hacer un programa en Pygame que muestre una ventana principal de juego con un fondo de color verde. La ventana debe tener un tamaño de 800x600 píxeles. Un ejemplo del programa funcionando puede ser como este:



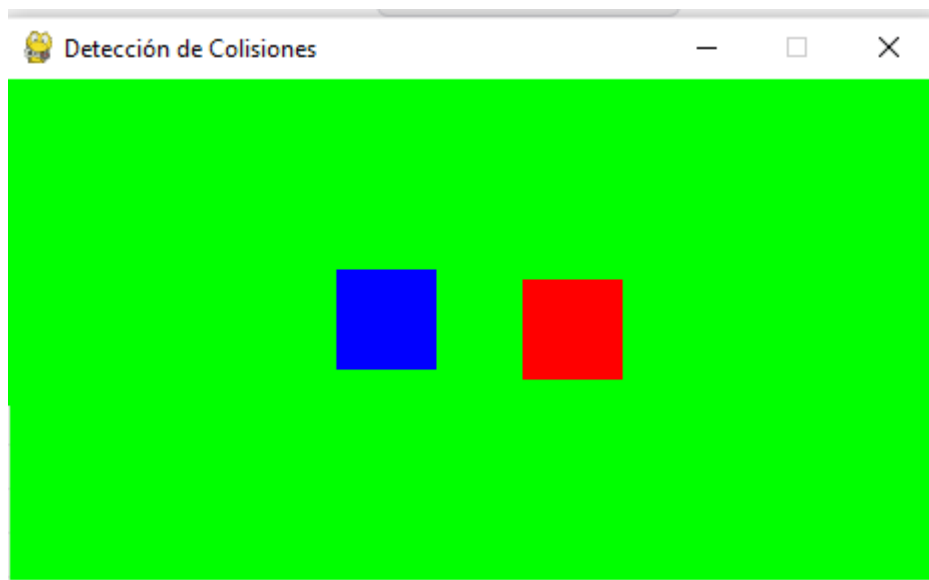
8. Hacer un programa en Pygame que muestre una ventana principal, dentro de la misma un objeto de tipo rectángulo y que muestre la cantidad de clic que hace el usuario. Un ejemplo del programa funcionando puede ser como este:



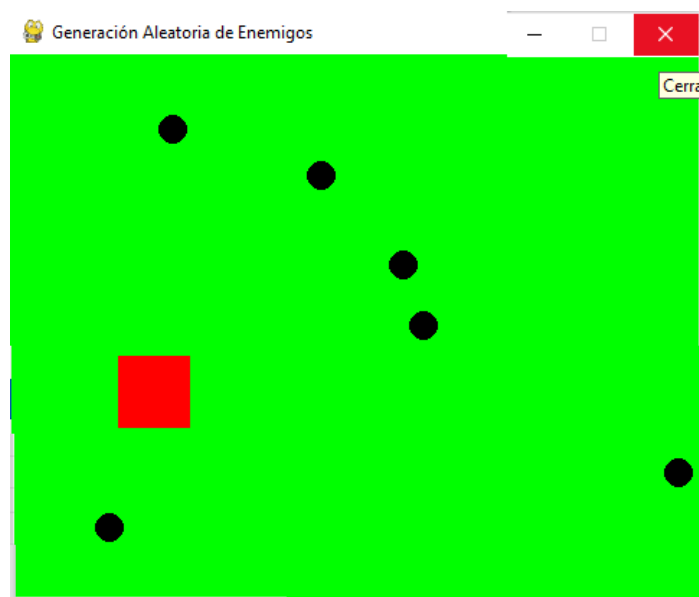
9. Hacer un programa en Pygame que muestre una ventana principal de color celeste, dentro de la misma un objeto de tipo rectángulo rojo y que se mueva en función del movimiento del teclado (flechas de dirección del cursor). Un ejemplo del programa funcionando puede ser como este:



10. Hacer un programa en Pygame que muestre una ventana principal de color verde o también puede introducir una imagen del tipo jpg como fondo, dentro de la misma, crear dos objetos de color azul y otro rojo. Los objetos se deben mover y el programa tiene que detectar colisiones. Un ejemplo del programa funcionando puede ser como este:



11 Hacer un programa en Pygame que muestre una ventana principal de color verde o también puede introducir una imagen del tipo jpg como fondo, dentro de la misma, crear un objeto principal (rectángulo rojo) que se mueva por teclado. Y crear en forma aleatoria otros objetos que se muestren al azar (villanos). El juego debe detectar las colisiones. Un ejemplo del programa funcionando puede ser como este:



### Preguntas de Revisión:

*Nota: Brindamos a modo de guía de estudio, los principales temas que abarca el desarrollo de la temática del módulo. El estudiante puede consultar bibliografía brindada por la catedra. '*

- a) ¿Qué es Tkinter y para qué se utiliza en Python?
- b) ¿Cómo se crea una ventana principal en Tkinter?
- c) ¿Qué son los widgets en Tkinter y puedes dar ejemplos de algunos widgets comunes?
- d) ¿Cómo se puede manejar un evento de clic en un botón en Tkinter? Proporciona un ejemplo.
- e) ¿Qué es el método pack() y cómo se diferencia de los métodos grid() y place() en Tkinter?
- f) ¿Cómo se puede personalizar la apariencia de un widget en Tkinter (por ejemplo, cambiar el color de fondo de un botón)? ¿Cómo se maneja un evento en Tkinter, por ejemplo, hacer clic en un botón?
- g) ¿Cómo puedes obtener el valor ingresado en un widget Entry?
- h) ¿Qué son los 'frames' en Tkinter y cómo se utilizan?
- i) ¿Qué es Pygame? ¿Qué características tiene?
- j) ¿Cómo se inicializa y se cierra correctamente una ventana de Pygame en un programa básico?
- k) ¿Qué estructura de datos en Pygame se utiliza comúnmente para manejar la posición y las dimensiones de objetos en pantalla, y cómo se puede detectar una colisión entre dos de estos objetos?
- l) ¿Cómo se maneja el evento de teclado en Pygame para permitir que un personaje se mueva en la pantalla?
- m) ¿Qué son las colisiones? ¿Cómo se administran desde Pygame?

### FIN TRABAJO PRACTICO N° 08