



FORMATO DE PLANEACIÓN

Estrategia didáctica


DATOS GENERALES

Nombre del participante	Denise Cruz Miranda
Asignatura	Matemáticas I-IV
Año o semestre en que imparte	2023
Horas clase a la semana	5 horas
Unidad	Matemáticas II: Unidad 1. Ecuaciones cuadráticas.
Aprendizajes	Aprendizaje: Identifica la naturaleza de las raíces de una ecuación cuadrática, a partir de sus coeficientes.
Problemática que se abordará a través del problema.	Realizar un programa que ayude a resolver ecuaciones de segundo grado, con el que identificaremos las raíces de la ecuación y la parábola que forma, utilizando la interfaz gráfica de Python.
Justificación. (porque considera que el programa en python o Julia puede apoyar al alumno a entender o lograr el aprendizaje)	<i>Que los estudiantes conozcan el método de Fórmula General para resolver ecuaciones de segundo grado, viendo paso a paso la solución, la gráfica de la ecuación, que distinga cuando la ecuación tenga raíces Reales y Complejas, y que observe las intersecciones con los ejes X y Y.</i>



Producto esperado (Después de haber explicado, haber realizado alguna actividad guiada y/o dejar una actividad extraclase, ¿Qué evidencia tiene que entregar para ser evaluada?)	Un programa que ayude a la resolución de ecuaciones de segundo grado dando las raíces, la parábola y la solución paso a paso.
Recursos materiales /Herramientas TIC	<ul style="list-style-type: none">• Computadora o laptop,• Software: simuladores, IDE (Entorno de desarrollo integrado) www.replit.com.• Conexión a internet.• Plataforma educativa: Moodle, Teams, Classroom, etc.• Videoprojector.• Pizarrón• Lista de cotejo para evaluar el desarrollo del proyecto, a manera de cronograma, para apoyar en el seguimiento del mismo.
Tiempos de realización.	2 clases de 4 horas y una clase de 1 hora.

Secuencia didáctica

	Presentación del problema a resolver
	Se requiere un programa que ayude a los alumnos a obtener las raíces y la parábola de una ecuación de segundo grado

Secuencia didáctica

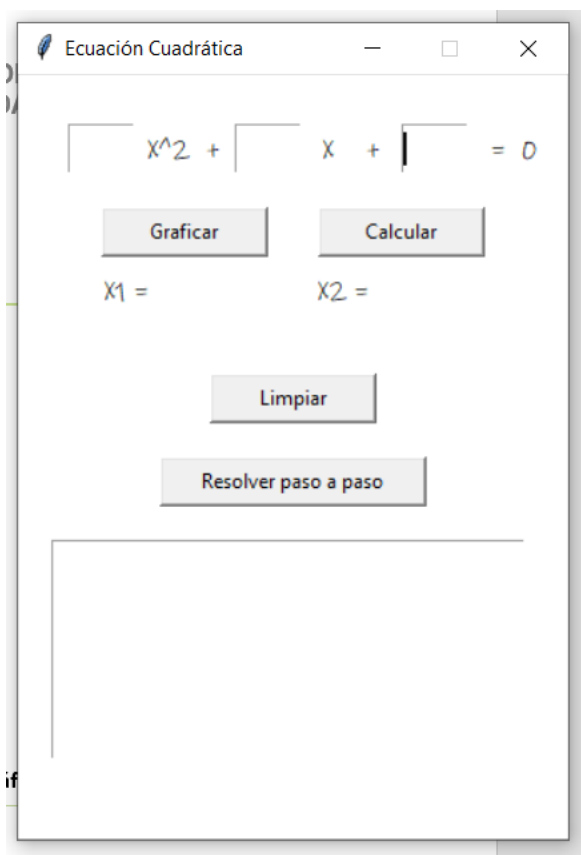


Figura 1. Interfaz gráfica



Inicio de la Sesión

Solicita a los alumnos que observen e identifiquen cuales son los elementos gráficos que encuentran en la ventana “Ecuación gráfica”. Comenta a los alumnos del grupo qué para el desarrollo del programa, el profesor explica el comportamiento de los elementos a utilizar:

- Button
- root = Tk()
- etiqueta1 = Label
- entrada1 = Entry(root)
- pasos_text = Text



Secuencia didáctica

- resultado = Label

Importar los paquetes que permitirán utilizar la ventana y el widget para el programa

- from tkinter import *
- from tkinter import ttk
- from io import StringIO



Desarrollo de la sesión

El profesor realiza una explicación acerca de las ecuaciones de segundo grado, propone algunos ejercicios para que los alumnos las resuelvan en equipos de 4 personas, una vez entendido un poco el tema se prenden las computadoras, se abre el entorno gráfico Replit y de forma guiada van metiendo cada uno de los datos para crear el programa y a su vez se va explicando cada paso para que los alumnos entiendan más acerca de la programación.

Conforme van entendiendo, se deja que los alumnos vayan completando el programa para crear botones, funciones, ventanas, etc.



Cierre de la sesión

Un ejercicio recapitulativo donde muestren su comprensión acerca de la programación y las ecuaciones de segundo grado, que cada equipo muestre su programa y se le pide que haga pruebas con diferentes ecuaciones.

Reflexionan en grupo acerca de lo que fue más fácil, más difícil y que hagan aportaciones individuales del trabajo hecho.



Evaluación

ACTIVIDAD A EVALUAR	PUNTAJE
Resolución de los ejercicios principales	2
Trabajo en equipo	2



Secuencia didáctica

	<table><tr><td>El código elaborado realiza la solución de ecuaciones de segundo grado</td><td>3</td></tr><tr><td>Entregaron en tiempo y forma</td><td></td></tr><tr><td>Asistencia</td><td>1</td></tr></table>	El código elaborado realiza la solución de ecuaciones de segundo grado	3	Entregaron en tiempo y forma		Asistencia	1
El código elaborado realiza la solución de ecuaciones de segundo grado	3						
Entregaron en tiempo y forma							
Asistencia	1						
	Evaluación Puede usar rúbrica, lista de cotejo, kahot, crusigrama, etc.						
	Referencias <ul style="list-style-type: none">• <i>Replit EduTools.</i> (s. f.). https://edutools.tec.mx/es/colecciones/tecnologias/replit• <i>¿Qué es Python? - Explicación del lenguaje Python - AWS.</i> (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes.• <i>El tutorial de Python.</i> (s. f.). Python documentation. https://docs.python.org/es/3/tutorial/index.html• <i>Aprende Python - Python España.</i> (s. f.). https://es.python.org/aprende-python/						



Secuencia didáctica

- Daniel Carreón. (2020, 1 marzo). *ECUACIONES DE SEGUNDO GRADO POR FORMULA GENERAL super facil -Para principiantes* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=ZC67c5ar9mA>

Anexo Códigos de resultados de ejercicios:

Propuesta 1:

```
In [1]: import math
import matplotlib.pyplot as plt
import numpy as np
```

```
In [2]: def ecuacion_segundo_grado(a, b, c):
discrim = b**2 - 4*a*c
if discrim > 0: #raíces reales diferentes
    x1 = (-b + math.sqrt(discrim)) / (2*a)
    x2 = (-b - math.sqrt(discrim)) / (2*a)
    return x1, x2
elif discrim == 0: #raíz reales repetidas
    x = -b / (2*a)
    return x, x
else: #raíz compleja
    parte_real = -b / (2*a)
    parte_imag = math.sqrt(abs(discriminante)) / (2*a)
    x1 = complex(parte_real, parte_imag)
    x2 = complex(parte_real, -parte_imag)
    return x1, x2
```

```
In [3]: def pasos_ecuacion(a, b, c):
print("Resolviendo la ecuación ax^2 + bx + c = 0")
print("Paso 1: Calculando el discriminante")
discrim = b**2 - 4*a*c
print("Discriminante = {discrim}")

if discrim > 0:
    print("Paso 2: El discriminante es mayor que 0, hay dos raíces reales y distintas")
    x1 = (-b + math.sqrt(discrim)) / (2*a)
    x2 = (-b - math.sqrt(discrim)) / (2*a)
    print("Las raíces son x1 = {x1} y x2 = {x2}")
elif discrim == 0:
    print("Paso 2: El discriminante es igual a 0, hay una raíz real repetida")
    x = -b / (2*a)
    print("La raíz repetida es x = {x}")
```



```
print("Las raíces son x1 = {x1} y x2 = {x2}")  
elif discriminante == 0:  
    print("Paso 2: El discriminante es igual a 0, hay una raíz real repetida")  
    x = -b / (2*a)  
    print("La raíz repetida es x = {x}")  
else:  
    print("Paso 2: El discriminante es menor que 0, hay dos raíces complejas (no reales)")  
    parte_real = -b / (2*a)  
    parte_imag = math.sqrt(abs(discriminante)) / (2*a)  
    x1 = complex(parte_real, parte_imag)  
    x2 = complex(parte_real, -parte_imag)  
    print("Las raíces complejas son x1 = {x1} y x2 = {x2}")
```

```
In [4]: # Introducir valores para a, b y c  
a = float(input("Ingrese el valor de a:"))  
b = float(input("Ingrese el valor de b:"))  
c = float(input("Ingrese el valor de c:"))
```

Ingrese el valor de a:1
Ingrese el valor de b:-9
Ingrese el valor de c:3

```
In [5]: # Solución paso a paso  
pasos_ecuacion(a, b, c)
```

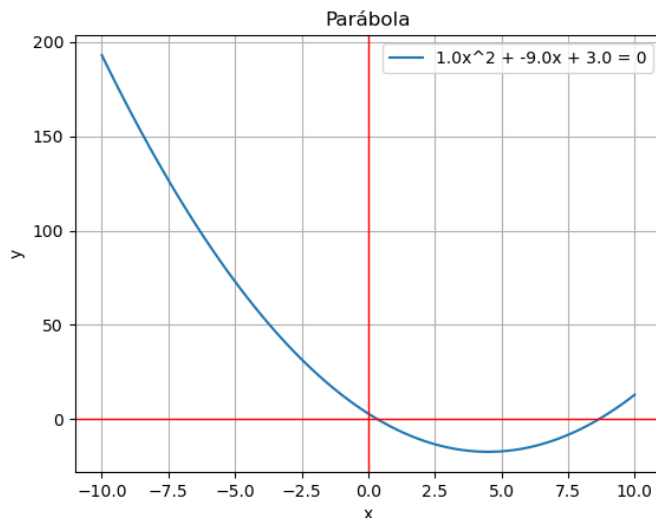
Resolviendo la ecuación $ax^2 + bx + c = 0$
Paso 1: Calculando el discriminante
Discriminante = {discriminante}
Paso 2: El discriminante es mayor que 0, hay dos raíces reales y distintas
Las raíces son x1 = {x1} y x2 = {x2}

```
In [6]: # Calcular las raíces  
x1, x2 = ecuacion_segundo_grado(a, b, c)  
print("Las raíces son:")  
print("x1 =", x1)  
print("x2 =", x2)
```

```
print("x1 =", x1)  
print("x2 =", x2)
```

Las raíces son:
x1 = 8.653311931459037
x2 = 0.34668806854096257

```
In [16]: # Graficar la parábola  
x = np.linspace(-10, 10, 100)  
y = a * x**2 + b * x + c  
  
plt.plot(x, y, label=f"{a}x^2 + {b}x + {c} = 0")  
plt.axhline(y=0, color='red', linewidth=1)  
plt.axvline(x=0, color='red', linewidth=1)  
plt.legend()  
plt.xlabel("x")  
plt.ylabel("y")  
plt.title("Parábola")  
plt.grid(True)  
plt.show()
```





Propuesta 2:

```
In [1]: from tkinter import *
        from tkinter import ttk

In [2]: import math
        import matplotlib.pyplot as plt
        import numpy as np
        from io import StringIO

In [3]: root = Tk()
        root.title("Ecuación Cuadrática")
        root.geometry("330x460")
        variable = StringVar()
        variable2 = StringVar()

In [4]: def ecuacion_segundo_grado():
        a = float(entrada1.get())
        b = float(entrada2.get())
        c = float(entrada3.get())

        discrim = (b*b)-4*a*c
        if discrim < 0:
            etiqueta = Label(root, text="No existen soluciones Reales", background="white")
            etiqueta.place(x=60, y=180, width=200, height=30)
            etiqueta.config(font=("Ink Free", 12))
        else:
            x1 = (-b + math.sqrt(discrim)) / (2 * a)
            x2 = (-b - math.sqrt(discrim)) / (2 * a)

            a = "{:.6f}".format(x1)
            b = "{:.6f}".format(x2)

            r = a, "    ", b

            print("X1 : ", "{:.6f}".format(x1))
            print("X2 : ", "{:.6f}".format(x2))

            return variable.set(r)

In [5]: def f(a, b, c, x):
        return a * (x ** 2) + b * x + c

In [6]: # Abre una ventana donde introducirá a, b y c
        def parabola():

In [6]: # Abre una ventana donde introducirá a, b y c
        def parabola():
            a = float(entrada1.get())
            b = float(entrada2.get())
            c = float(entrada3.get())

            x = range(-100, 100)
            plt.plot(x, [f(a, b, c, i) for i in x])

            plt.axhline(0, color="red")
            plt.axvline(0, color="red")

            plt.xlim(-60, 60)
            plt.ylim(-60, 60)
            plt.draw()
            plt.xlabel("x")
            plt.ylabel("y")
            plt.grid(True)

            plt.title("Parábola de la ecuación de segundo grado")

            plt.show()

In [7]: # Limpia Las entradas y Los resultados
        def borrar_ventana():
            entrada1.delete(0, END)
            entrada2.delete(0, END)
            entrada3.delete(0, END)
            variable.set("")
            variable2.set("")
            pasos_text.delete("1.0", "end")
            root.update_idletasks()

In [8]: # Resuelve paso a paso La ecuación por el método de La Fórmula General.
        def pasos_ecuacion():

            try:
                a = float(entrada1.get())
                b = float(entrada2.get())
                c = float(entrada3.get())

                pasos_text.delete("1.0", END)
                pasos_text.insert(END, "Resolviendo la ecuación ax^2 + bx + c = 0\n")
                pasos_text.insert(END, "Calculando el discriminante.\n")
```




```
pasos_text.delete(1.0, END)
pasos_text.insert(END, "Resolviendo la ecuación  $ax^2 + bx + c = 0$ \n")
pasos_text.insert(END, "Calculando el discriminante.\n")

discrim = b**2 - 4*a*c

if discrim < 0:
    pasos_text.insert(END, "El discriminante es menor que 0, hay dos raíces complejas, es decir, no tocan al eje X.\n")
    parte_real = -b / (2*a)
    parte_imag = math.sqrt(abs(discrim)) / (2*a)
    x1 = complex(parte_real, parte_imag)
    x2 = complex(parte_real, -parte_imag)
    pasos_text.insert(END, "Raíz 1:  $x_1 = \{x1:.6f\}$ \n")
    pasos_text.insert(END, "Raíz 2:  $x_2 = \{x2:.6f\}$ \n")
else:
    pasos_text.insert(END, "Calculando las raíces reales.\n")
    x1 = (-b + math.sqrt(discrim)) / (2 * a)
    x2 = (-b - math.sqrt(discrim)) / (2 * a)
    pasos_text.insert(END, "Raíz 1:  $x_1 = \{x1:.6f\}$ \n")
    pasos_text.insert(END, "Raíz 2:  $x_2 = \{x2:.6f\}$ \n")

if discrim == 0:
    pasos_text.insert(END, "El discriminante es igual a 0, hay raíces reales repetidas, es decir, tocan el mismo punto.\n")
    pasos_text.insert(END, "Raíces repetidas:  $x_1x_2 = \{x1:.6f\}$ ")

except ValueError:
    pasos_text.delete(1.0, END)
    pasos_text.insert(END, "Por favor, ingrese valores numéricos válidos para a, b y c.")
```

In [9]: #Entradas

```
entrada1 = Entry(root)
entrada1.place(x = 30, y = 30, width = 40, height = 30)
entrada1.config(font = ("Ink Free",12))

entrada2 = Entry(root)
entrada2.place(x = 130, y = 30, width = 40, height = 30)
entrada2.config(font = ("Ink Free",12))

entrada3 = Entry(root)
entrada3.place(x = 230, y = 30, width = 40, height = 30)
entrada3.config(font = ("Ink Free",12))
```

In [10]: # Etiquetas

```
etiqueta1 = Label(root, text = "X^2 +", background="white")
etiqueta1.place(x = 75, y = 30, width = 50, height = 30)
etiqueta1.config(font = ("Ink Free",12))

etiqueta2 = Label(root, text = "X +", background="white")
etiqueta2.place(x = 175, y = 30, width = 50, height = 30)
etiqueta2.config(font = ("Ink Free",12))

etiqueta3 = Label(root, text = "= 0", background="white")
etiqueta3.place(x = 270, y = 30, width = 50, height = 30)
etiqueta3.config(font = ("Ink Free",12))
```

In [11]: # Botones

```
boton1 = Button(root, text = "Graficar", command = parabola)
boton1.place(x = 50, y = 80, width = 100, height = 30)

boton2 = Button(root, text = "Calcular", command = ecuacion_segundo_grado)
boton2.place(x = 180, y = 80, width = 100, height = 30)

boton3 = Button(root, text="Limpiar", command= borrar_ventana)
boton3.place(x=115, y=180, width=100, height=30)

boton4 = Button(root, text="Resolver paso a paso", command = pasos_ecuacion)
boton4.place(x=85, y=230, width=160, height=30)
```

In [12]: # Widget para mostrar Los pasos

```
pasos_text = Text(root, wrap=WORD, width=35, height=8)
pasos_text.place(x=20, y=280)
```

In [13]: # Resultado

```
resultado = Label(root, text = "X1 =", background = "white")
resultado.place(x = 50, y = 120, width = 30, height = 20)
resultado.config(font = ("Ink Free",12))

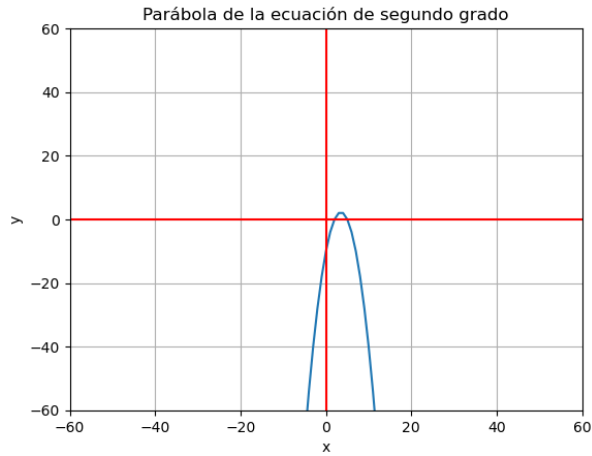
resultado = Label(root, text = "X2 =", background = "white")
resultado.place(x = 180, y = 120, width = 30, height = 20)
resultado.config(font = ("Ink Free",12))

resultado1 = Label(root, textvariable = variable, background = "white")
resultado1.place(x = 50, y = 140, width = 230, height = 30)
resultado1.config(font = ("Ink Free",12))
```

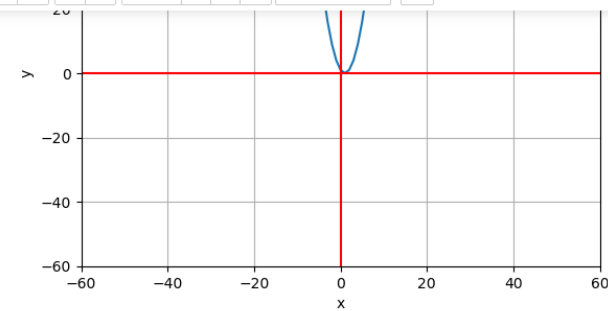
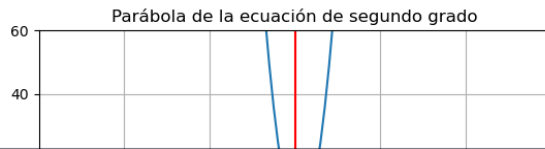


In [14]: # Configuración de la ventana

```
root.resizable(0,0)  
root.config(bg = "white", cursor = "target")  
root.mainloop()
```



X1 : 2.000000
X2 : 5.000000



X1 : 1.000000
X2 : 1.000000

