**FORMATO DE PLANEACIÓN   
Estrategia didáctica**

DATOS GENERALES

|  |  |
| --- | --- |
| Nombre del participante | Jeanett Figueroa Martínez |
| Asignatura | Cibernética y Computación II |
| Año o semestre en que imparte | Sexto semestre |
| Horas clase a la semana | 4 horas |
| Unidad | Unidad IV Interfaz gráfica de usuario |
| Aprendizajes | Elabora programas con una interfaz gráfica de usuario, aplicando las Clases: JFrame, JLabel y JButton |

|  |  |
| --- | --- |
| **Problemática que se abordará a través del problema.** | Realizar una calculadora con operaciones aritméticas básicas (suma, resta, multiplicación y división) utilizando la interfaz gráfica en específico Jframe, Jlabel, JButton |
| **Justificación.**  **(porque considera que el programa en python o Julia puede apoyar al alumno a entender o lograr el aprendizaje)** | El programa en Python muestra la similitud del ejercicio elaborado en el lenguaje Java, apoya a entender el uso de las etiquetas, los botones y las acciones dentro de cada botón de forma creativa, utilizando la lógica matemática, yendo desde el análisis de como realiza el proceso una calculadora, para después interpretar en un código de lenguaje y refuerza el trabajo colaborativo. |
| **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?** | Una calculadora que realice las operaciones aritméticas básicas de suma, resta, multiplicación y división, en modo gráfico. |
| **Recursos materiales /Herramientas TIC** | * Computadora o laptop, * Software: Lenguaje de programación Python * Conexión a internet, * Plataforma educativa: Moodle * Videoproyector, * Pizarrón, * Plumigis * Lista de cotejo para evaluar el desarrollo del proyecto, a manera de cronograma, para apoyar en el seguimiento del mismo. |
| **Tiempos de realización** | 8 horas 4 en clase y 2 extra clase |
| **Forma de trabajo** | En equipos de forma colaborativa |

| Secuencia didáctica | |
| --- | --- |
|  | **Presentación del problema a resolver** |
| **SE requiere:** Una calculadora que realice las operaciones aritméticas básicas de suma, resta, multiplicación y división, en modo gráfico.    **Figura 1. Calculadora elaborada en una interfaz gráfica** |
|
|  | **Inicio de la Sesión** |
| Solicita a los alumnos que observen e identifiquen cuales son los elementos gráficos que encuentran en la figura 1.  Comenta a los alumnos del grupo qué para el desarrollo de la Calculadora, el profesor explica el comportamiento de los elementos a utilizar:    Button  grid  display.get  display.delete  display.insert  Importar los paquetes que permitirán utilizar los widgets para la calculadora.  from tkinter import \*  from tkinter import messagebox |
|  | **Desarrollo de la sesión** |
| El profesor realiza de forma guiada la entrada de datos de la calculadora, y un botón de operación y el botón = encargado de mostrar el resultado .  Se conforman equipos de cuatro alumnos  A cada integrante le tocará programar los botones que restan |
|  | **Cierre de la sesión** |
| Cada equipo muestra su resultado  Reflexionan en grupo, el profesor realiza las siguientes preguntas:  ¿Qué se le dificulto?  ¿Qué fue lo más fácil?  ¿Cuál fue el aporte individual al trabajar en equipo? |
|  | **Evaluación** |
| |  |  | | --- | --- | | Actividad a evaluar | Puntaje | | Trabajo en equipo (aporto el trabajo correspondiente) | 1 | | Lograron llegar a acuerdos para que la tarea fuera fácil (comunicación, organización). | 2 | | El código elaborado realiza la operación correspondiente a la tarea designada | 2 | | Entregó el equipo en tiempo y forma | 2 | |
|  | **Referencias** |
| Alvarez, A. (2016). Guía de tkinter. <https://buildmedia.readthedocs.org/media/pdf/guia-tkinter/latest/guia-tkinter.pdf>  Stac Overflow contribuitors. (s/f). Aprendizaje tkinter Ebook gratis. <https://riptutorial.com/Download/tkinter-es.pdf> |

Anexo. Código resultado del ejercicio:

# programa en python que abre una ventana la que permite realizar operaciones aritméticas básicas simulando una calculadora

from tkinter import \*

from tkinter import messagebox

class Pycalc(Frame):

# inicializa la calculadora

def \_\_init\_\_(self, master, \*args, \*\*kwargs):

Frame.\_\_init\_\_(self, master, \*args, \*\*kwargs)

self.parent = master

self.grid()

self.createWidgets()

# borra el caracter

def deleteLastCharacter(self):

textLength = len(self.display.get())

if textLength >= 1:

self.display.delete(textLength - 1, END)

if textLength == 1:

self.replaceText("0")

def replaceText(self, text):

self.display.delete(0, END)

self.display.insert(0, text)

# visualización de la entrada de los datos

def append(self, text):

actualText = self.display.get()

textLength = len(actualText)

if actualText == "0":

self.replaceText(text)

else:

self.display.insert(textLength, text)

# evalua si existe algún error en la entrada de la operación a realizar

def evaluate(self):

try:

self.replaceText(eval(self.display.get()))

except (SyntaxError, AttributeError):

messagebox.showerror("Error", "Error de sintaxis")

self.replaceText("0")

except ZeroDivisionError:

messagebox.showerror("Error", "No se puede dividir entre 0")

self.replaceText("0")

# dependiendo del signo realiza la operación

def containsSigns(self):

operatorList = ["\*", "/", "+", "-"]

display = self.display.get()

for c in display:

if c in operatorList:

return True

return False

def changeSign(self):

if self.containsSigns():

self.evaluate()

firstChar = self.display.get()[0]

if firstChar == "0":

pass

elif firstChar == "-":

self.display.delete(0)

else:

self.display.insert(0, "-")

# realiza la operacion incersa

def inverse(self):

self.display.insert(0, "1/(")

self.append(")")

self.evaluate()

# crea la tabla con la apariencia de una calculadora

def createWidgets(self):

self.display = Entry(self, font=("Arial", 24), relief=RAISED, justify=RIGHT, bg='darkblue', fg='red', borderwidth=0)

self.display.insert(0, "0")

self.display.grid(row=0, column=0, columnspan=4, sticky="nsew")

self.ceButton = Button(self, font=("Arial", 12), fg='red', text="CE", highlightbackground='red', command=lambda: self.replaceText("0"))

self.ceButton.grid(row=1, column=0, sticky="nsew")

self.inverseButton = Button(self, font=("Arial", 12), fg='red', text="1/x", highlightbackground='lightgrey', command=lambda: self.inverse())

self.inverseButton.grid(row=1, column=2, sticky="nsew")

self.delButton = Button(self, font=("Arial", 12), fg='red', text="Del", highlightbackground='red', command=lambda: self.deleteLastCharacter())

self.delButton.grid(row=1, column=1, sticky="nsew")

self.divButton = Button(self, font=("Arial", 12), fg='red', text="/", highlightbackground='lightgrey', command=lambda: self.append("/"))

self.divButton.grid(row=1, column=3, sticky="nsew")

self.sevenButton = Button(self, font=("Arial", 12), fg='red', text="7", highlightbackground='black', command=lambda: self.append("7"))

self.sevenButton.grid(row=2, column=0, sticky="nsew")

self.eightButton = Button(self, font=("Arial", 12), fg='red', text="8", highlightbackground='black', command=lambda: self.append("8"))

self.eightButton.grid(row=2, column=1, sticky="nsew")

self.nineButton = Button(self, font=("Arial", 12), fg='red', text="9", highlightbackground='black', command=lambda: self.append("9"))

self.nineButton.grid(row=2, column=2, sticky="nsew")

self.multButton = Button(self, font=("Arial", 12), fg='red', text="\*", highlightbackground='lightgrey', command=lambda: self.append("\*"))

self.multButton.grid(row=2, column=3, sticky="nsew")

self.fourButton = Button(self, font=("Arial", 12), fg='red', text="4", highlightbackground='black', command=lambda: self.append("4"))

self.fourButton.grid(row=3, column=0, sticky="nsew")

self.fiveButton = Button(self, font=("Arial", 12), fg='red', text="5", highlightbackground='black', command=lambda: self.append("5"))

self.fiveButton.grid(row=3, column=1, sticky="nsew")

self.sixButton = Button(self, font=("Arial", 12), fg='red', text="6", highlightbackground='black', command=lambda: self.append("6"))

self.sixButton.grid(row=3, column=2, sticky="nsew")

self.minusButton = Button(self, font=("Arial", 12), fg='red', text="-", highlightbackground='lightgrey', command=lambda: self.append("-"))

self.minusButton.grid(row=3, column=3, sticky="nsew")

self.oneButton = Button(self, font=("Arial", 12), fg='red', text="1", highlightbackground='black', command=lambda: self.append("1"))

self.oneButton.grid(row=4, column=0, sticky="nsew")

self.twoButton = Button(self, font=("Arial", 12), fg='red', text="2", highlightbackground='black', command=lambda: self.append("2"))

self.twoButton.grid(row=4, column=1, sticky="nsew")

self.threeButton = Button(self, font=("Arial", 12), fg='red', text="3", highlightbackground='black', command=lambda: self.append("3"))

self.threeButton.grid(row=4, column=2, sticky="nsew")

self.plusButton = Button(self, font=("Arial", 12), fg='red', text="+", highlightbackground='lightgrey', command=lambda: self.append("+"))

self.plusButton.grid(row=4, column=3, sticky="nsew")

self.negToggleButton = Button(self, font=("Arial", 12), fg='red', text="+/-", highlightbackground='lightgrey', command=lambda: self.changeSign())

self.negToggleButton.grid(row=5, column=0, sticky="nsew")

self.zeroButton = Button(self, font=("Arial", 12), fg='red', text="0", highlightbackground='black', command=lambda: self.append("0"))

self.zeroButton.grid(row=5, column=1, sticky="nsew")

self.decimalButton = Button(self, font=("Arial", 12), fg='red', text=".", highlightbackground='lightgrey', command=lambda: self.append("."))

self.decimalButton.grid(row=5, column=2, sticky="nsew")

self.equalsButton = Button(self, font=("Arial", 12), fg='red', text="=", highlightbackground='lightgrey', command=lambda: self.evaluate())

self.equalsButton.grid(row=5, column=3, sticky="nsew")

Calculator = Tk()

Calculator.title("Calculadora con Python")

Calculator.resizable(False, False)

Calculator.config(cursor="arrow")

root = Pycalc(Calculator).grid()

Calculator.mainloop()