

Tecnológico Nacional de México

Instituto Tecnológico de Celaya

Examen de la materia Lenguajes y autómatas II

Fecha 30/01/24

Nombres: Víctor Fernando Sánchez Alvarado
Alexandro León Bolaños

1.- Construye un autómata que pueda reconocer un número en formato entero, hexadecimal o flotante. Tome los siguientes ejemplos como algunos casos válidos:

- a) 1234
- b) 23.22214
- c) 3.45e-23
- d) 0x43fef
- e) -3456.3442e+12
- f) -2134
- g) 0xAAAAF1

Si construyes el autómata real con POO, la calificación es hasta 100 puntos.

Si construyes el autómata solo simulado con programación la calificación máxima es 80.

Código

```
import re
import tkinter as tk
from tkinter import messagebox

class NumeroReconocedor:
    def __init__(self):
        self.patron = r'^(\+|-)?(0x[a-fA-F0-9]+|(\d+(\.\d+)?)?([eE](\+|-)?\d+)?)$'

    def es_valido(self, numero):
        return re.match(self.patron, numero) is not None

class Aplicacion:
    def __init__(self, root):
        self.root = root
        root.title("Reconocedor de Números")
        self.set_window_size(240, 120)

        self.reconocedor = NumeroReconocedor()

        self.label = tk.Label(root, text="Introduce un número:")
        self.label.pack()

        self.entrada = tk.Entry(root)
        self.entrada.pack()

        self.boton = tk.Button(root, text="Verificar", command=self.verificar)
        self.boton.pack()

        self.info_label = tk.Label(root, text="Formatos válidos:\nEntero, Decimal,\nHexadecimal, Científico\n(ej. 123, 23.45, 0x1A3, -4.5e+7)", justify=tk.LEFT)
        self.info_label.pack()

    def verificar(self):
```

```
numero = self.entrada.get().replace(' ', '') # Eliminar espacios
if not numero:
    messagebox.showwarning("Advertencia", "Por favor, introduce un número.")
    return
if self.reconocedor.es_valido(numero):
    messagebox.showinfo("Resultado", f"'{numero}' es un número válido.")
else:
    messagebox.showwarning("Resultado", f"'{numero}' no es un número válido.")

def set_window_size(self, width, height):
    screen_width = self.root.winfo_screenwidth()
    screen_height = self.root.winfo_screenheight()
    x = (screen_width - width) // 2
    y = (screen_height - height) // 2
    self.root.geometry(f'{width}x{height}+{x}+{y}')

root = tk.Tk()
app = Aplicacion(root)
root.mainloop()
```