

```

# Archivo main.py

import webbrowser
import time
from ClassMenu import Menus

class ElblogdeMia:
    """
    ¿Qué es @staticmethod?

    @staticmethod es un decorador que le dice a Python:
    "Este método NO usa self ni la instancia.
    Trátalo como una función normal que vive dentro de la clase."

    Es decir:
    - No recibe self
    - No recibe cls
    - No depende de atributos de la clase
    - No necesita que crees un objeto para usarlo
    """

    @staticmethod
    def main():
        """Programa principal"""

        menu = Menus()

        while True:
            menu.menu_principal()
            tema = input("-> ").lower()

            seleccionados = {
                "p": "Programación",
                "c": "Cocina",
                "h": "Historia",
                "l": "Logística"
            }

            if tema in seleccionados:
                print(f"Selección: {seleccionados[tema]} – un buen tema de
estudio")
                time.sleep(2)

            elif tema == "e":
                webbrowser.open("https://danielcaraballo.es/")
                break

            else:
                print("Opción no válida")
                time.sleep(2)
                continue

        menus = {

```

```

        "p": menu.menu_programacion,
        "c": menu.menu_cocina,
        "h": menu.menu_historia,
        "l": menu.menu_logistica
    }

    if tema in menus:
        menus[tema]()

if __name__ == "__main__":
    ElblogdeMia.main()

#_____

# Archivo ClassMenu.py

"""Clase Menús"""

from os import system
import webbrowser
import threading
import time

class Menus:
    """Clase que gestiona los distintos menús del blog de Daniel."""

    def menu_principal(self):
        """Muestra el menú principal."""
        system("cls")
        print("""
        En este script puedes seleccionar temas del Blog de Daniel

        p -> Programación
        h -> Historia
        l -> Logística
        c -> Cocina
        e -> Salir
        """)

    def menu_programacion(self):
        """Menú de selección de lenguajes de programación."""

        while True:
            system("cls")
            print("""
            Selecciona lenguaje de programación:
            1. Python
            2. C (Hola Mundo)
            3. Java
            4. C#
            5. Mojo
            """)

```

```

        6. Regresar al menú anterior
    """
    programa = input("-> ")

    opciones = {
        "1":
"https://danielcaraballo.es/lenguaje-de-programacion-python/",
        "2":
"https://danielcaraballo.es/entendiendo-el-clasico-hola-mundo-en-c/",
        "3": "https://danielcaraballo.es/5646-2/",
        "4":
"https://danielcaraballo.es/creacion-de-aplicaciones-de-escritorio-con-c/#google_vignette",
        "5":
"https://danielcaraballo.es/category/lenguajes-de-programacion/mojo/"
    }

    if programa in opciones:
        webbrowser.open_new_tab(opciones[programa])
    elif programa == "6":
        break
    else:
        print("Opción no disponible")
        time.sleep(2)

def menu_cocina(self):
    """Abre la sección de cocina en una nueva pestaña."""
    threading.Thread(
        target=lambda: webbrowser.open(
            "https://danielcaraballo.es/category/cocina/#google_vignette"
        )
    ).start()
    time.sleep(2)

def menu_historia(self):
    """Abre la sección de historia."""
    webbrowser.open_new_tab("https://danielcaraballo.es/category/historia/")
    time.sleep(2)

def menu_logistica(self):
    """Abre la sección de logística."""

webbrowser.open_new_tab("https://danielcaraballo.es/category/articulos/")
time.sleep(2)

```

#
