

Código de la app

Melissa Macedo Ramos

February 2025

1 código

```
#code final
import random
import pandas as pd
import pulp

# Función para ingresar cuántos días tendrá el horario
def ingresar_dias_del_horario():
    while True:
        try:
            dias = int(input("¿Cuántos días tendrá tu horario? (1-5 días): "))
            if 1 <= dias <= 5:
                return dias
            else:
                print("Por favor, ingrese un número entre 1 y 5.")
        except ValueError:
            print("Entrada inválida. Por favor ingrese un número entero.")

# Función para seleccionar el turno (mañana o tarde)
def seleccionar_turno():
    while True:
        turno = input("Seleccione el turno (mañana/tarde): ").strip().lower()
        if turno in ["mañana", "tarde"]:
            return turno
        else:
            print("Por favor, ingrese 'mañana' o 'tarde'.")

# Función para asignar los horarios según el turno seleccionado
def asignar_horarios_por_turno(dias_semana, turno):
    if turno == "mañana":
        horario_entrada = 7 # 7:00 AM
        horario_salida = 13 # 1:00 PM
    elif turno == "tarde":
        horario_entrada = 13 # 1:00 PM
```

```

        horario_salida = 19 # 7:00 PM

    horarios_por_dia = {}
    for dia in dias_semana:
        horarios_por_dia[dia] = (horario_entrada, 0, horario_salida, 0)

    return horarios_por_dia

# Función para ingresar la cantidad de cursos
def ingresar_cantidad_de_cursos():
    while True:
        try:
            cantidad = int(input("¿Cuántos cursos desea ingresar? "))
            if cantidad > 0:
                return cantidad
            else:
                print("Por favor ingrese un número mayor que 0.")
        except ValueError:
            print("Entrada inválida. Por favor ingrese un número entero.")

# Función para ingresar los nombres de los cursos
def ingresar_nombres_de_cursos(cantidad_cursos):
    cursos = []
    for i in range(cantidad_cursos):
        curso = input(f"Ingrese el nombre del curso {i+1}: ")
        cursos.append(curso)
    return cursos

# Función para ingresar las horas semanales por cada curso
def ingresar_horas_por_curso(cursos):
    horas_por_curso = {}
    for curso in cursos:
        while True:
            try:
                horas = int(input(f"Ingrese la cantidad de horas semanales para el curso {curso}: "))
                if horas > 0:
                    horas_por_curso[curso] = horas
                    break
            except:
                print("Por favor ingrese un número mayor que 0.")
        except ValueError:
            print("Entrada inválida. Por favor ingrese un número entero.")
    return horas_por_curso

# Función para ingresar el número de opciones a generar
def ingresar_numero_de_opciones():

```

```

while True:
    try:
        opciones = int(input("¿Cuántas opciones de horarios desea generar? "))
        if opciones > 0:
            return opciones
        else:
            print("Por favor ingrese un número mayor que 0.")
    except ValueError:
        print("Entrada inválida. Por favor ingrese un número entero.")

# Definir los días de la semana
dias_semana_completa = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes"]

# Función para ingresar las horas máximas por curso por día
def ingresar_max_horas_por_curso_por_dia():
    while True:
        try:
            max_horas = int(input("¿Cuántas horas máximas por día puede llevar un curso? (Ej: 6) "))
            if max_horas > 0:
                return max_horas
            else:
                print("Por favor ingrese un número mayor que 0.")
        except ValueError:
            print("Entrada inválida. Por favor ingrese un número entero.")

# Modificación en la función optimizar_horario
def optimizar_horario(horas_por_materia, dias_semana, max_horas_por_dia=6, max_horas_por_curso=6):
    # Crear el problema de optimización
    prob = pulp.LpProblem("Optimización_de_Horarios", pulp.LpMinimize)

    # Crear las variables de decisión (horas asignadas por curso en cada día)
    variables = {}
    for curso in horas_por_materia:
        for dia in dias_semana:
            variables[(curso, dia)] = pulp.LpVariable(f"{curso}_{dia}", lowBound=0, cat='Integer')

    # Función objetivo: minimizar las horas no asignadas o desbalanceadas
    prob += pulp.lpSum([variables[(curso, dia)] for curso in horas_por_materia for dia in dias_semana])

    # Restricciones: Cada curso debe tener asignado exactamente su número de horas semanales
    for curso, horas in horas_por_materia.items():
        prob += pulp.lpSum([variables[(curso, dia)] for dia in dias_semana]) == horas

    # Restricción: Las horas asignadas en cada día no deben exceder las horas máximas por día
    for dia in dias_semana:
        prob += pulp.lpSum([variables[(curso, dia)] for curso in horas_por_materia]) <= max_horas_por_dia

```

```

# Restricción: Las horas por curso en cada día no deben exceder el máximo de 3 horas por
for curso in horas_por_materia:
    for dia in dias_semana:
        prob += variables[(curso, dia)] <= max_horas_por_curso_por_dia

# Resolver el problema
prob.solve()

# Convertir el horario optimizado en un formato de horario visual
horario_optimizado = {dia: [] for dia in dias_semana}
for curso in horas_por_materia:
    for dia in dias_semana:
        horas_asignadas = pulp.value(variables[(curso, dia)])
        if horas_asignadas > 0:
            for _ in range(int(horas_asignadas)):
                horario_optimizado[dia].append(curso)

# Agregar aleatoriedad si se solicita
if aleatoriedad:
    for dia in dias_semana:
        random.shuffle(horario_optimizado[dia]) # Mezcla los cursos de cada día aleatoriamente

# Crear DataFrame similar al código base
horario_df = pd.DataFrame.from_dict(horario_optimizado, orient='index').fillna('')

return horario_df

# Función principal para generar y mostrar los horarios
def generar_y_mostrar_horarios():
    dias_totales = ingresar_dias_del_horario()
    dias_semana = dias_semana_completa[:dias_totales]
    turno = seleccionar_turno()
    horarios_por_dia = asignar_horarios_por_turno(dias_semana, turno)
    cantidad_cursos = ingresar_cantidad_de_cursos()
    cursos = ingresar_nombres_de_cursos(cantidad_cursos)
    horas_por_materia = ingresar_horas_por_curso(cursos)
    num_opciones = ingresar_numero_de_opciones()

    # Ingresar las horas máximas por curso por día
    max_horas_por_curso_por_dia = ingresar_max_horas_por_curso_por_dia()

    for i in range(num_opciones):
        # Introducimos aleatoriedad en la generación de opciones
        aleatoriedad = i > 0 # Para la primera opción no hay aleatoriedad, para las demás sí
        horario_optimizado = optimizar_horario(

```

```
        horas_por_materia, dias_semana, max_horas_por_curso_por_dia=max_horas_por_curso,
print(f"Horario Opción {i+1}")
print(horario_optimizado)
print("\n" + "-"*50 + "\n")

# Llamar a la función para generar y mostrar los horarios
generar_y_mostrar_horarios()e
```