UNIVERSIDAD DEL VALLE DE GUATEMALA

CAMPUS CENTRAL

ALGORITMOS Y PROGRAMACIÓN BÁSICA

ASESOR: ERICK MARROQUIN

FECHA DE ENTREGA: 03/08/2025

PROYECTO – FASE 2

PROTOTIPADO

MARÍA FERNANDA AFRE CÓRDOVA 251355

LAURA GABRIELA SANDOVAL PALMA 251409

MARÍA ALEJANDRA AFRE CÓRDOVA 25135

Parte 1 – Prototipo funcional

1. **Implementen un prototipo funcional basado en su idea final.**

```
from datetime import datetime, timedelta
# Entradas del usuario
materias = []
num_materias = int(input("¿Cuántas materias estás cursando? "))

for _ in range(num_materias):
    nombre = input("\nNombre de la materia: ")
    dificultad = input("Dificultad (alta/media/baja): ").lower()
    entrega_str = input("Fecha de entrega más próxima (formato YYYY-MM-DD): ")
    entrega = datetime.strptime(entrega_str, "%Y-%m-%d")
    materias.append({"nombre": nombre, "dificultad": dificultad, "entrega": entrega})

# Energía por hora (7 AM a 10 PM)
print("\nIngresa tu nivel de energía de 1 a 10 para cada hora:")
energia_horas = {}
for h in range(7, 22):
    nivel = int(input(f"{h}:00 - Nivel de energía: "))
    energia_horas[h] = nivel

# Horarios disponibles para estudiar
horas_disponibles = []
```

```
print("\nIndica tus horas disponibles para estudiar (ej. 9 para 9:00 AM). Escribe 'fin' para terminar:")

while True:

    entrada = input("Hora disponible: ")

    if entrada.lower() == 'fin':

        break

    horas_disponibles.append(int(entrada))


# Proceso: asignar horas según dificultad y energía

print("\n📅 Horario sugerido de estudio:\n")


materias_ordenadas = sorted(materias, key=lambda m: (m['entrega'], {"alta": 3, "media": 2, "baja": 1}[m['dificultad']]), reverse=False)


plan_estudio = {}


for materia in materias_ordenadas:

    bloques_asignados = []

    dificultad = materia['dificultad']

    bloques_necesarios = {"alta": 3, "media": 2, "baja": 1}[dificultad]


    # Ordenar las horas disponibles por nivel de energía

    horas_ordenadas = sorted(horas_disponibles, key=lambda h: energia_horas[h], reverse=True)
```

```
        for hora in horas_ordenadas:
            if hora not in plan_estudio and len(bloques_asignados) < bloques_necesarios:
                plan_estudio[hora] = materia['nombre']
                bloques_asignados.append(hora)

    # Salida: mostrar el plan ordenado por hora
    for hora in sorted(plan_estudio):
        print(f"{hora}:00 - {plan_estudio[hora]}")
```

2. **El prototipo debe resolver el problema definido en la Fase 1.**

Versión inicial

- El programa pedía ingresar materias, dificultad, fecha próxima de entrega, horas disponibles y nivel de energía por hora.
- A partir de esa información, generaba un plan de estudio ordenado y asignado por nivel de prioridad.

Parte 2 – Validación con usuarios

1. **Realicen al menos 5 sesiones de validación con usuarios reales.**

https://drive.google.com/drive/folders/16hWOv1jrLW-Dn9ptMnZPmqdz4hLae4-7?usp=sharing

2. **Documenten las observaciones y hallazgos.**

Logramos observar que el código se tornaba un poco tedioso al momento de preguntar sobre las materias, ya que los usuarios llegaron a confundirse al momento de ingresar las horas o se volvía

tedioso tener que volver a ingresar los mismos datos para todas las materias, también pudo tornarse confuso el formato de fechas que indicaba el código enviar, también el código solamente preguntaba por una fecha próxima de entrega en lugar de centrarse en la carga que podrían tener algunos cursos en los que tengan entregas de proyectos, tareas o parciales.

Por lo que a partir de estas observaciones decidimos implementar mejoras al código para que pudiese ser más especifico sin ser tan tedioso y ser más claro.

3. **Identifiquen las mejoras necesarias a partir de la retroalimentación.**

hacer que después de preguntarle al usuario cuantas tareas tiene le pregunte cuales desea estudiar, luego hacer que en lugar de la fecha entrega pregunte cuantas tareas o parciales próximos tiene de la materia si el número es =0 no preguntar fecha de entrega y si es mayor a 0 preguntar la fecha de entrega por tarea, luego que antes de las horas de energía pregunte cuantas horas al dia tiene disponibles lunes, martes, miércoles ..., de qué hora a qué hora ejemplo de 9-12 (para indicar de 9 a.m. a 12 p.m.)

4. **Implementen las mejoras en su prototipo.**

Versión mejorada (tras validación con usuarios)

- El sistema ahora permite registrar múltiples entregas por materia (tareas y parciales).
- Se ajustó para preguntar horarios disponibles por día de la semana en lugar de una lista general.
- Se mantienen las preferencias de energía, pero con un flujo más claro.
- La asignación de bloques de estudio es más flexible y personalizada.

```
from datetime import datetime
from collections import defaultdict
```

```
    materias = []
    dias_semana = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]

    # Paso 1: Materias que desea estudiar
    num_materias = int(input("¿Cuántas materias deseas incluir en tu horario de estudio? "))

    for _ in range(num_materias):
        nombre = input("\nNombre de la materia: ")
        dificultad = input("Dificultad (alta/media/baja): ").lower()

        entregas = []
        cantidad_tareas = int(input(f"¿Cuántas tareas o parciales próximos tienes en {nombre}?
"))
        if cantidad_tareas > 0:
            for i in range(cantidad_tareas):
                fecha_str = input(f" - Fecha de entrega #{i+1} (formato YYYY-MM-DD): ")
                fecha = datetime.strptime(fecha_str, "%Y-%m-%d")
                entregas.append(fecha)

        materias.append({
            "nombre": nombre,
            "dificultad": dificultad,
            "entregas": entregas
        })

    # Paso 2: Horarios disponibles por día
    horas_disponibles = defaultdict(list)

    print("\nIndica tus horarios disponibles por día (formato 9-12 para 9:00 a 12:00). Puedes
ingresar varios rangos por día.")
    for dia in dias_semana:
        print(f"\n{dia.title()}:")
```

```
    while True:
        rango = input(" - Rango disponible (o escribe 'fin' para continuar): ")
        if rango.lower() == "fin":
            break
        try:
            inicio, fin = map(int, rango.split("-"))
            for h in range(inicio, fin):
                horas_disponibles[dia].append(h)
        except:
            print("❌ Formato incorrecto. Usa formato como 9-12.")

# Paso 3: Energía por hora
energia_horas = {}
print("\nIngresa tu nivel de energía por hora (1 a 10, solo entre 7 AM y 10 PM):")
for h in range(7, 22):
    nivel = int(input(f"{h}:00 - Nivel de energía: "))
    energia_horas[h] = nivel

# Paso 4: Asignar bloques de estudio
plan_estudio = defaultdict(list)

# Ordenar materias por fecha más próxima y dificultad
materias_ordenadas = sorted(
    materias,
    key=lambda m: (min(m["entregas"]) if m["entregas"] else datetime.max,
                   {"alta": 3, "media": 2, "baja": 1}[m["dificultad"]])
)

print("\n🗓️ Plan de estudio sugerido por día:\n")

for dia in dias_semana:
```

```
        horas_dia = horas_disponibles[dia]
        horas_ordenadas = sorted(horas_dia, key=lambda h: energia_horas.get(h, 0), reverse=True)
        asignadas = 0

        for materia in materias_ordenadas:
            bloques_necesarios = {"alta": 3, "media": 2, "baja": 1}[materia["dificultad"]]
            bloques_asignados = 0

            for h in horas_ordenadas:
                if all(h != hora for hora, _ in plan_estudio[dia]) and bloques_asignados < bloques_necesarios:
                    plan_estudio[dia].append((h, materia["nombre"]))
                    bloques_asignados += 1
                    asignadas += 1

    # Paso 5: Mostrar horario sugerido
    for dia in dias_semana:
        print(f"\n{dia.title()}:")
        if not plan_estudio[dia]:
            print("  - No se asignaron sesiones de estudio.")
        else:
            for h, materia in sorted(plan_estudio[dia]):
                print(f"  {h}:00 - {materia}").
```

Parte 3 – Retrospectiva del trabajo en equipo (5 pts)

Retrospectiva – Estrella de Mar

1. Seguir Haciendo

Lo que nos funcionó bien queremos mantener:

- Definir claramente el problema.

- Utilizar las entrevistas y perfiles de usuario para identificar las necesidades reales.
- Identificar los datos de entrada, proceso, salida para llevar un proceso simple y funcional..
- Priorizar la funcionalidad mínima viable para validar la idea rápidamente.

2. Empezar a Hacer

Ideas o prácticas que podríamos incorporar en el futuro.

- Poner la opción de exportar el horario en formato PDF o integrarlo con Google Calendar.

3. Más de

Las acciones que fueron útiles y podrían hacerse más:

- Validar decisiones con los perfiles de usuario obtenidos (entrevistas o encuestas).
- Hacer más pruebas con distintos perfiles (procrastinador, trabajador, matutino).
- Investigar buenas prácticas para estudiantes.

4. Menos de

Las cosas que se hicieron, pero podrían simplificarse:

- Pedir entradas manuales demasiado largas (como energía hora por hora) podria simplificarse con una escala general por bloque (mañana/tarde/noche).
- Evitar repetir preguntas si no son necesarias
- Preguntar solo sobre las materias que se desea estudiar
- Especificar si la fecha de entrega es de parciales o tareas

5. Dejar de Hacer

Prácticas que no funcionaron o ya no aportan valor:.

- Asumir que todos los estudiantes usan el sistema igual: cada perfil necesita recomendaciones distintas.