Solucionarios

Semana 3

07.07.2025 - 13.07.2025

Caso General

Sistema de Gestión de Tareas

Organizar las tareas es fundamental tanto en la vida personal como en la profesional. Un sistema de gestión de tareas permite llevar un registro de lo que hay que hacer, marcar el progreso y priorizar el trabajo. La lógica detrás de una simple "To-Do List" es la base de software complejo como Trello, Asana o Jira.

- Yamil Ayma

Nivel Fácil: Agregar una Nueva Tarea

```python
def agregar_tarea(lista_tareas, descripcion):
    # 1. Creamos un diccionario para representar la nueva tarea.
    #    Tendrá la descripción que nos pasan y el estado 'completada' en False por defecto.
    nueva_tarea = {
        'descripcion': descripcion,
        'completada': False
    }

    # 2. Usamos el método .append() para añadir el nuevo diccionario al final de la lista.
    lista_tareas.append(nueva_tarea)

    # 3. Devolvemos la lista, que ahora contiene la nueva tarea.
    return lista_tareas

# --- Pruebas ---
def probar_agregar_tarea():
    # Prueba 1: Agregar a una lista vacía
    tareas = []
    tareas_actualizadas = agregar_tarea(tareas, "Estudiar Python")
    print(f"Prueba 1: {tareas_actualizadas == [{'descripcion': 'Estudiar Python', 'completada':
False}]}")

    # Prueba 2: Agregar a una lista existente
    tareas_existentes = [{'descripcion': 'Hacer ejercicio', 'completada': True}]
    tareas_actualizadas_2 = agregar_tarea(tareas_existentes, "Llamar al dentista")
    esperado = [
        {'descripcion': 'Hacer ejercicio', 'completada': True},
        {'descripcion': 'Llamar al dentista', 'completada': False}
    ]
    print(f"Prueba 2: {tareas_actualizadas_2 == esperado}")

probar_agregar_tarea()
```

★ Click para acceder al código ★

Nivel Medio: Marcar Tarea como Completada y Filtrar por Estado

```python
def marcar_y_filtrar(lista_tareas, descripcion, estado_filtro):
    """
    Marca una tarea como completada y luego filtra la lista según un estado.
    """
    # --- PASO 1: MARCAR LA TAREA COMO COMPLETADA ---
    for tarea in lista_tareas:
        # Si la descripción de la tarea actual coincide con la que buscamos
        if tarea['descripcion'] == descripcion:
            # Cambiamos el estado de 'completada'.
            tarea['completada'] = True
            # Usamos 'break' para detener la búsqueda una vez que la encontramos.
            break

    # --- PASO 2: FILTRAR LA LISTA SEGÚN EL ESTADO ---
    # Creamos una lista vacía para guardar los resultados del filtro.
    tareas_filtradas = []
    for tarea in lista_tareas:
        if estado_filtro == 'todas':
            tareas_filtradas.append(tarea)
        elif estado_filtro == 'pendientes' and not tarea['completada']:
            tareas_filtradas.append(tarea)
        elif estado_filtro == 'completadas' and tarea['completada']:
            tareas_filtradas.append(tarea)

    # Devolvemos la nueva lista que solo contiene las tareas filtradas.
    return tareas_filtradas

# --- Pruebas ---
def probar_marcar_y_filtrar():
    lista_base = [
        {'descripcion': 'Comprar leche', 'completada': False},
        {'descripcion': 'Pagar facturas', 'completada': False},
        {'descripcion': 'Sacar al perro', 'completada': False}
    ]

    tareas_p1 = [t.copy() for t in lista_base]
    resultado1 = marcar_y_filtrar(tareas_p1, 'Pagar facturas', 'completadas')
    print(f"Prueba 1: {resultado1 == [{'descripcion': 'Pagar facturas', 'completada': True}]}")

    tareas_p2 = [t.copy() for t in lista_base]
    resultado2 = marcar_y_filtrar(tareas_p2, 'Comprar leche', 'pendientes')
    esperado2 = [
        {'descripcion': 'Pagar facturas', 'completada': False},
        {'descripcion': 'Sacar al perro', 'completada': False}
    ]
    print(f"Prueba 2: {resultado2 == esperado2}")

    tareas_p3 = [t.copy() for t in lista_base]
    resultado3 = marcar_y_filtrar(tareas_p3, 'Leer un libro', 'todas')
    print(f"Prueba 3: {resultado3 == lista_base}")

probar_marcar_y_filtrar()
```

Click para acceder al código

Éxitos en los ejercicios

La simplicidad no precede a la complejidad, pero la sigue.
- Alan Perlis. Epigramas de la Programación

Aprende mucho