

DAM1 DAW1 / Entornos de desarrollo

Desarrollo de software

1.5 Practicamos



Autor: Lisa ERIKSEN

Fecha: 2025 / 2026

Crea tu propio Tablero Kanban en Python

Aprender a aplicar los principios de **Kanban** programando un pequeño sistema que gestione tareas según su estado:

"Por hacer", "En progreso" y "Hecho".

Paso 1: Crea la estructura del tablero

👉 Primero, crea un diccionario llamado **kanban** que contenga tres listas. Usa un **diccionario** con tres claves (Por hacer, En progreso, Hecho), y cada una con una lista vacía.

```
kanban = {  
    # Completa aquí con tres claves y listas vacías [ ]  
}
```

👉 ¿Cuál sería la ventaja de usar un diccionario con listas para representar el tablero?

Paso 2: Añade una función para agregar tareas

👉 Crea una función llamada **agregar_tarea(nombre)** que añada una nueva tarea a la lista "Por hacer". Usa `append()` para añadir la tarea a la lista correspondiente.

```
def agregar_tarea(nombre):  
    # Completa aquí para añadir el parámetro 'nombre' a la lista kanban["Por hacer"]  
    print(f"Tarea '{nombre}' añadida a 'Por hacer'.")
```

👉 Prueba tu código llamando a la función con una tarea, por ejemplo:

```
agregar_tarea("Diseñar interfaz")
```

Paso 3: Crea una función para mover tareas

Crea una función **mover_tarea(nombre, nuevo_estado)** que cambie una tarea de una columna a otra.

```
def mover_tarea(nombre, nuevo_estado):
    for estado, tareas in kanban.items():
        if nombre in tareas:
            # Completa aquí para eliminar la tarea 'nombre' de la lista 'tareas'
            # Completa aquí para añadir la tarea 'nombre' en el nuevo estado kanban[nuevo_estado]
            print(f"Tarea '{nombre}' movida a '{nuevo_estado}'.")
            return
    print("Tarea no encontrada.")
```

👉 ¿Por qué usamos un bucle for dentro de esta función?

Paso 4: Muestra el estado del tablero

👉 Crea una función **mostrar_tablero()** que imprima todas las tareas según su estado.

```
def mostrar_tablero():
    print("\n Estado actual del tablero:")
    # Completa aquí con un bucle for para recorrer todos los elementos del diccionario kanban
    print(f"{estado}: {tareas}")
```

👉 Prueba tu código:

```
agregar_tarea("Crear base de datos")
mover_tarea("Diseñar interfaz", "En progreso")
mover_tarea("Diseñar interfaz", "Hecho")
```

Desafío extra

Agrega una función que cuente cuántas tareas hay en cada columna.

Pista: usa `len(tareas)` dentro de un bucle.