

# UD03 – PE (Prueba Escrita) - 01

## Control de Tareas Pendientes en Python

**Asignatura:** (Optativa) Programación Multiplataforma en Python

**Unidad Didáctica:** Mod 03 - Toma de decisiones, bucles y listas

**Puntuación total: 10 puntos**

### Objetivo del ejercicio

Aplicar de forma integrada los conocimientos de las secciones:

- 3.1 – Toma de decisiones en Python (uso de if, elif, else)
- 3.2 – Bucles en Python (uso de while y for)
- 3.4 a 3.7 – Listas, operaciones y aplicaciones avanzadas

El estudiante desarrollará un programa que permita **gestionar una lista de tareas pendientes**, utilizando estructuras de datos, control de flujo y funciones en Python.

### Descripción general

Desarrolla un programa en **Python** que permita gestionar un **control de tareas pendientes**, aplicando el uso de **listas, condicionales y bucles**.

Cada tarea debe representarse como una **lista con tres elementos**:

tarea = [nombre, prioridad, estado]

donde:

- nombre → Cadena de texto con el nombre de la tarea.
- prioridad → Número entero del 1 al 10 (1 = baja, 10 = alta).
- estado → Cadena con el valor "pendiente" o "finalizada".

### Objetivos de aprendizaje

- Practicar el uso de **listas de listas**.
- Utilizar **condicionales** (if, elif, else).
- Aplicar **bucles** para recorrer listas.
- Manipular listas con los métodos y técnicas de ordenamiento.
- Implementar un **menú interactivo** con opciones múltiples.
- Nota:

**Se entrega archivo Python. Devolver dicho archivo y realizar el ejercicio en él.**

## Apartado 1 (1 punto): Menú principal

Crea un menú que muestre las siguientes opciones:

1. Agregar tarea
2. Cambiar estado de tarea
3. Eliminar tarea
4. Ordenar tareas por prioridad
5. Buscar tarea
6. Mostrar todas las tareas
7. Listar tareas por prioridad específica
8. Salir

El menú debe estar dentro de un **bucle while** y usar **condicionales if-elif-else**.

## Apartado 2 (2 puntos): Agregar tarea

Crea una función `agregar_tarea(tareas)` que:

- Solicite al usuario el nombre y la prioridad (validando que esté entre 1 y 10).
- Cree la lista con elementos de la forma [nombre, prioridad, estado].
- Agregue la tarea a la lista principal.

## Apartado 3 (2 puntos): Cambiar estado y eliminar tareas

1. `cambiar_estado(tareas, nombre, estado)`
  - Cambia el estado de una tarea de "pendiente" a "finalizada" o viceversa.
2. `eliminar_tarea(tareas, nombre)`
  - Elimina una tarea según su nombre.
  - Muestra un mensaje por consola si la tarea no existe.

## Apartado 4 (2 puntos): Ordenar tareas por prioridad

Implementa una función `ordenar_tareas_por_prioridad(tareas)` que devuelva **una copia de la lista de tareas ordenada de mayor a menor prioridad**. Puedes usar el **ordenamiento de burbuja** (**se suministra y hay que adaptarlo**).

## Apartado 5 (1.5 puntos): Buscar tarea

Crea una función `buscar_tarea(tareas)` que permita:

- Pedir el nombre de una tarea.
- Mostrar su información si existe.
- Mostrar un mensaje por consola si no se encuentra.

## Apartado 6 (1 punto): Mostrar todas las tareas

Diseña mostrar\_tareas(tareas) que muestre las tareas registradas con el siguiente formato:

1. Estudiar Python | Prioridad: 8 | Estado: pendiente  
Etc...

## Apartado 7 (0.5 puntos): Listar tareas por prioridad

Crea una función listar\_tareas\_por\_prioridad(tareas) que:

- Solicite al usuario una prioridad específica (por ejemplo, 8).
- Muestre **solo las tareas con esa prioridad**.
- Si no hay ninguna, muestre un mensaje por consola indicándolo.

## Distribución de puntuación

Apartado	Descripción	Puntos
1	Menú principal (bucle y condicionales)	1
2	Agregar tarea	2
3	Cambiar estado y eliminar tarea	2
4	Ordenar tareas por prioridad	2
5	Buscar tarea	1.5
6	Mostrar todas las tareas	1
7	Listar tareas por prioridad	0.5
<b>TOTAL</b>		<b>10 puntos</b>

### Nota: (para todos los ejercicios)

Responder exactamente a lo que se pregunta y como se pregunta.

No usar IA. No usar Internet.

Ejercicio obligatorio e individual de cada alumno.

No se puede usar ningún material no suministrado por el profesor.

El código debe de estar libre de errores.

Realizar comentarios y documentar tu solución.

Formatear el código para lectura clara e indentación correcta que facilite la lectura del código.

Generar y justificar tu solución aportada.

Prohibido el uso del móvil en el aula y durante el examen.

Permanecer en silencio durante el examen (no se habla con el compañero).

Entregar dicho ejercicio en la plataforma siguiendo instrucciones del profesor, tiene hora de entrega (si se entrega fuera de hora se considera no entregado).

Leer atentamente todas las preguntas.