

Working Time 😎

L6: Sentencias iterativas

Es hora de que pongas en práctica todo lo aprendido. 😎

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las soluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

¡Manos a la obra!

1. Desafío 🎯

“Analizador de temperaturas (versión modular)”

Construí un programa que lea temperaturas ($^{\circ}\text{C}$) y calcule estadísticas usando **bucles** (`for`, `while`) y **separando la lógica en funciones**. Debe ofrecer **dos modos** de carga y producir un **reporte** con estadísticas y una salida “visual”.

Requisitos funcionales (sí o sí con funciones):

1. **Modo A:** `for` + `range` con **cantidad fija** de días.
2. **Modo B:** `while` con **centinela** (escribí `fin` para terminar).
3. **Validaciones** con `while`: reintentar si el dato es inválido.
4. **Cálculos iterativos:** **mínimo**, **máximo** y **promedio** recorriendo la lista con `for` (evitá `min()`, `max()`, `sum()`).
5. En **Modo A**, hacé un **segundo recorrido** para contar cuántas temperaturas están **por encima del promedio**.
6. **Salida “visual”:** imprimí una barra por temperatura usando repetición de caracteres, p.ej. '#' * n.

2. ¿Dónde se lleva a cabo? 💻

Crea un **cuaderno Jupyter** llamado `Leccion_6_Iterativas.ipynb` y resuelve **todo dentro del .ipynb** (no `.py`).

Sugerencia de estructura de tu .ipynb:

- **Portada** (Markdown): título + tu nombre.

- **Consigna** (Markdown): copiá este enunciado.
- **Desarrollo** (Code): código con funciones y bucles.
- **Pruebas** (Code): al menos **3 escenarios** distintos.
- **Conclusiones** (Markdown): qué aprendiste sobre **for, while**, contadores y validaciones.

3. Tiempo de dedicación

1 Hora.

4. Recursos

[Documentación de clases y objetos en Python](#)

[Video explicativo sobre OOP en Python](#)

5. Plus

- **break y continue**: en Modo B, salteá líneas vacías con **continue**.
- **for-else**: avisá si **no** hubo ninguna temperatura > 35 °C.
- **Ordenar sin sorted()**: implementá un **bubble sort** simple con **for** anidados para mostrar la lista ordenada.
- **Módulo externo propio**: separá las funciones en un archivo **util_temperaturas.py** e importalo (sin librerías externas).

6. Condición

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante puedes compartir tus resultados con el resto de los bootcampers y construir conocimiento en conjunto.

7. Resolución del ejercicio:

Mostrar la resolución de la actividad (con paso a paso, capturas de pantalla, etc.) para que el estudiante pueda hacer una comparación de sus resultados con lo esperado.

Celda 1 — Título (Markdown)

```
# Lección 6 – Sentencias iterativas (Python)
## Resolución del desafío: Analizador de temperaturas (versión modular)
Aplicamos `for`, `while`, centinelas y validaciones, organizando la solución en funciones.
```

Celda 2 — Selección de modo y orquestación

```
def elegir_modo():
    print("== Analizador de temperaturas ==")
    print("A) for + range con cantidad fija")
    print("B) while con 'fin' (centinela)")
    while True:
        modo = input("Modo (A/B): ").strip().upper()
        if modo in ("A", "B"):
            return modo
        print("Opción inválida. Elegí A o B.")

def main():
    modo = elegir_modo()
    if modo == "A":
        temperaturas = leer_temperaturas_modo_a()
    else:
        temperaturas = leer_temperaturas_modo_b()

    if len(temperaturas) == 0:
        print("No se cargaron temperaturas.")
        return

    minimo, maximo, promedio = calcular_min_max_prom(temperaturas)

    print("\n== RESULTADOS ==")
    print(f"Cantidad: {len(temperaturas)}")
    print(f"Mínimo: {minimo:.2f} °C")
    print(f" Máximo: {maximo:.2f} °C")
```

```

print(f"Promedio: {promedio:.2f} °C")

if modo == "A":
    sobre_prom = contar_mayores_que_promedio(temperaturas,
promedio)
    print(f"Por encima del promedio: {sobre_prom}")

dibujar_barras(temperaturas)

```

Celda 3 — Lectura Modo A (for + range) con validación

```

def leer_temperaturas_modo_a():
    # Validar n con while (entero > 0)
    while True:
        n_str = input("¿Cuántos días vas a cargar? ")
        if n_str.isdigit() and int(n_str) > 0:
            n = int(n_str)
            break
        print("Valor inválido. Ingresá un entero mayor que 0.")

    # Carga con for + range
    temps = []
    for i in range(n):
        cargado = False
        while not cargado:
            s = input(f"Temperatura día {i+1}: ").strip()
            try:
                t = float(s.replace(",", "."))
                temps.append(t)
                cargado = True
            except ValueError:
                print("Entrada inválida. Ingresá un número (ej: 23.4).")
    return temps

```

Celda 4 — Lectura Modo B (while + centinela fin)

```

def leer_temperaturas_modo_b():
    print("Ingresá temperaturas una por línea. Escribí 'fin' para terminar.")
    temps = []
    while True:
        s = input("Temperatura (o 'fin'): ").strip().lower()
        if s == "fin":

```

```

        break
    if s == "": # plus opcional: saltar líneas vacías
        continue
    try:
        t = float(s.replace(",","."))
        temps.append(t)
    except ValueError:
        print("Entrada inválida. Ingresá un número o 'fin'.")
return temps

```

Celda 5 — Estadísticas iterativas (sin min/max/sum)

```

def calcular_min_max_prom(temperaturas):
    minimo = temperaturas[0]
    maximo = temperaturas[0]
    total = 0.0
    for t in temperaturas:
        total += t
        if t < minimo:
            minimo = t
        if t > maximo:
            maximo = t
    promedio = total / len(temperaturas)
    return minimo, maximo, promedio

def contar_mayores_que_promedio(temperaturas, promedio):
    cuenta = 0
    for t in temperaturas:
        if t > promedio:
            cuenta += 1
    return cuenta

```

Celda 6 — Salida “visual” con barras

```

def dibujar_barras(temperaturas):
    print("\n==== Barras por temperatura ====")
    # for con range para índice + repetición de caracteres
    for i in range(len(temperaturas)):
        t = temperaturas[i]
        cantidad = int(round(t)) if t > 0 else 0
        print(f"Día {i+1:>2}: {'#' * cantidad} {t:.2f}°C")

```

Celda 7 — Ejecutar el programa

```
# Ejecutá esta celda para correr el flujo completo
if __name__ == "__main__":
    main()
```