

## Estatus



- **Nombre:** Jesús Eduardo Jiménez Covarrubias
- **Materia:** Computación Tolerante a Fallas
- **Profesor:** Michel Emanuel López Franco
- **NRC:** 179961
- **Sección:** D06
- **Ciclo:** 2025-A

## Instrucciones:

Realizar un programa que sea capaz de revisar el estado de tu aplicación.

<https://python-para-impacientes.blogspot.com/2016/12/threading-programacion-con-hilos-i.html>

<https://tecnobillo.com/sections/python-en-windows/servicios-windows-python/servicios-windows-python.html>

<https://stackoverflow.com/questions/3044580/multiprocessing-vs-threading-python>

## Desarrollo:

### Código (Python):

```
import threading
import time
import psutil
import socket
import platform # Nuevo: Para detectar el sistema operativo
from flask import Flask
app = Flask(__name__)

# Verificación de servicios solo en Windows
SISTEMA_WINDOWS = platform.system() == "Windows"
if SISTEMA_WINDOWS:
    try:
        import win32serviceutil
    except ImportError:
        print("\nAdvertencia: pywin32 no instalado. Verificación de servicios deshabilitada.")
        win32serviceutil = None
else:
    win32serviceutil = None

class MonitorThread(threading.Thread):
    def __init__(self, proceso_nombre, servicio_nombre=None, puerto=None, intervalo=5):
        super().__init__()
        self.proceso_nombre = proceso_nombre
        self.servicio_nombre = servicio_nombre
        self.puerto = puerto
        self.intervalo = intervalo
        self.detener = threading.Event()
        self.lock = threading.Lock()

    def verificar_proceso(self):
        for proc in psutil.process_iter(['name']):
            if proc.info['name'] == self.proceso_nombre:
                return True
        return False

    def verificar_servicio_windows(self):
        if not SISTEMA_WINDOWS or not self.servicio_nombre:
            return "No aplicable"
```

```

        if not win32serviceutil:
            return "Requiere pywin32"

        try:
            status =
win32serviceutil.QueryServiceStatus(self.servicio_nombre)
            return "Ejecutando" if status[1] == 4 else "Detenido"
        except Exception as e:
            return f"Error: {str(e)}"

    def verificar_puerto(self):
        if not self.puerto:
            return "No configurado"

        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(1)
        try:
            sock.connect(('localhost', self.puerto))
            return "Abierto"
        except (ConnectionRefusedError, socket.timeout):
            return "Cerrado"
        finally:
            sock.close()

    def run(self):
        while not self.detener.is_set():
            with self.lock:
                print("\n--- Estado de la Aplicación ---")
                print(f"Proceso '{self.proceso_nombre}': {'✅ Activo' if
self.verificar_proceso() else '❌ Inactivo'}")

                if self.servicio_nombre:
                    print(f"Servicio '{self.servicio_nombre}':
{self.verificar_servicio_windows()}")

                print(f"Puerto {self.puerto}: {self.verificar_puerto()}")
                print("-----")

            time.sleep(self.intervalo)

    def parar(self):
        self.detener.set()

if __name__ == "__main__":
    # CONFIGURACIÓN PERSONALIZABLE (¡Edita estos valores!)
    monitoreador = MonitorThread(
        proceso_nombre="python.exe", # Ej: "chrome.exe", "mysqld.exe"
        servicio_nombre="W3SVC", # Ej: "MySQL", solo para Windows
        puerto=5000, # Ej: 8080, 3306
        intervalo=3 # Segundos entre verificaciones
    )

    try:
        print("Iniciando monitoreo (Presiona Ctrl+C para detener)...")

```

```

        monitoreador.start()
        monitoreador.join()
    except KeyboardInterrupt:
        monitoreador.parar()
        print("\nMonitoreo detenido.")

@app.route('/')
def hello():
    return ";Servidor activo!"

if __name__ == '__main__':
    app.run(port=5000)

```

## Funcionamiento del Código:

El código proporcionado es un script de Python que combina dos funcionalidades principales:

1. **\*\*Un monitor de sistema\*\*** que verifica periódicamente el estado de un proceso, un servicio (solo en Windows) y un puerto.
2. **\*\*Un servidor web básico\*\*** usando Flask que muestra un mensaje simple.

Sin embargo, hay un **\*\*error crítico de diseño\*\*** que impide que ambas funcionalidades funcionen simultáneamente. Aquí está el análisis detallado:

**## \*\*Parte 1: Monitor de Sistema (MonitorThread)\*\***

**### \*\*Características:\*\***

- **\*\*Verificación de Proceso:\*\*** Usa `psutil` para comprobar si un proceso específico está en ejecución (ej. `python.exe`).
- **\*\*Verificación de Servicio (Windows):\*\*** Utiliza `win32serviceutil` para revisar el estado de un servicio de Windows (ej. `W3SVC` - servicio IIS).
- **\*\*Verificación de Puerto:\*\*** Intenta conectarse a un puerto local para determinar si está abierto.
- **\*\*Ejecución en Hilo:\*\*** Corre en segundo plano con un intervalo configurable.

**### \*\*Configuración:\*\***

```

monitoreador = MonitorThread(
    proceso_nombre="python.exe",
    servicio_nombre="W3SVC",

```

```

    puerto=5000,
    intervalo=3
)

```

### ## \*\*Parte 2: Servidor Web con Flask\*\*

- **Ruta ``/``:** Devuelve el mensaje "¡Servidor activo!".
- **Puerto 5000:** El servidor se ejecuta en el puerto 5000, el mismo que el monitor verifica.

### ## \*\*Problema Principal: Estructura del Código\*\*

El script tiene dos bloques ``if __name__ == '__main__':``:

1. **Primer Bloque:** Inicia el hilo del monitor y lo espera (``join()``), lo cual bloquea la ejecución.
2. **Segundo Bloque:** Inicia el servidor Flask, pero **nunca se alcanza** porque el primer bloque está esperando indefinidamente.

#### # Primer bloque (monitor)

```

if __name__ == "__main__":
    monitoreador.start()
    monitoreador.join() # ¡Bloquea aquí!

```

#### # Segundo bloque (Flask)

```

if __name__ == '__main__':
    app.run(port=5000) # Nunca se ejecuta

```

### ## \*\*Solución Propuesta\*\*

Para que ambas partes funcionen concurrentemente:

1. **Eliminar el ``join()``** del hilo del monitor.
2. **Unificar los bloques ``__main__``** para iniciar ambos componentes.

```

if __name__ == "__main__":
    # Iniciar monitor en segundo plano

```

```
monitoreador = MonitorThread(...)
monitoreador.start()
```

```
# Iniciar Flask en el hilo principal
app.run(port=5000)
```

## **\*\*Explicación Detallada del Código\*\***

### **\*\*1. Importaciones y Configuración Inicial\*\***

- **`psutil`**: Para manejar procesos.
- **`win32serviceutil`**: Solo en Windows, para gestionar servicios.
- **`Flask`**: Framework web.

### **\*\*2. Clase MonitorThread\*\***

- **Métodos de Verificación**:
  - `verificar_proceso()`: Busca el proceso por nombre.
  - `verificar_servicio_windows()`: Usa APIs de Windows.
  - `verificar_puerto()`: Intenta una conexión TCP.
- **Ejecución Continua**: En un bucle con `time.sleep()`.

### **\*\*3. Servidor Flask\*\***

- Ruta básica para confirmar que el servidor está activo.

## **\*\*Cómo Ejecutar Correctamente\*\***

1. **Instalar Dependencias**:

```
pip install flask psutil pywin32 (si estás en Windows)
```

2. **Modificar el Código** para corregir el error de estructura.

3. **Ejecutar el Script**: Ambos componentes (monitor y servidor) funcionarán en paralelo.

## **\*\*Posibles Mejoras\*\***

- **Interfaz Web**: Mostrar el estado del sistema via Flask.
- **Manejo de Errores**: Para casos como puertos ya en uso.

- **\*\*Compatibilidad Multiplataforma:\*\*** Mejorar la lógica de servicios para Linux/macOS.

## Servicio Activo

```

Seleccionar Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\jusue> cd "G:\Septimo Semestre\COMPUTACION TOLERANTE A FALLAS\Ejercicios\Ejercicio 6"
PS G:\Septimo Semestre\COMPUTACION TOLERANTE A FALLAS\Ejercicios\Ejercicio 6> pip install psutil
>>
>> # Si usas Windows y necesitas monitorear servicios:
>> pip install pywin32
Requirement already satisfied: psutil in c:\users\jusue\appdata\local\programs\python\python312\lib\site-packages (7.0.0)

[notice] A new release of pip is available: 24.2 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: pywin32 in c:\users\jusue\appdata\local\programs\python\python312\lib\site-packages (308)

[notice] A new release of pip is available: 24.2 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS G:\Septimo Semestre\COMPUTACION TOLERANTE A FALLAS\Ejercicios\Ejercicio 6> python Estatus.py
Iniciando monitoreo (Presiona Ctrl+C para detener)...

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

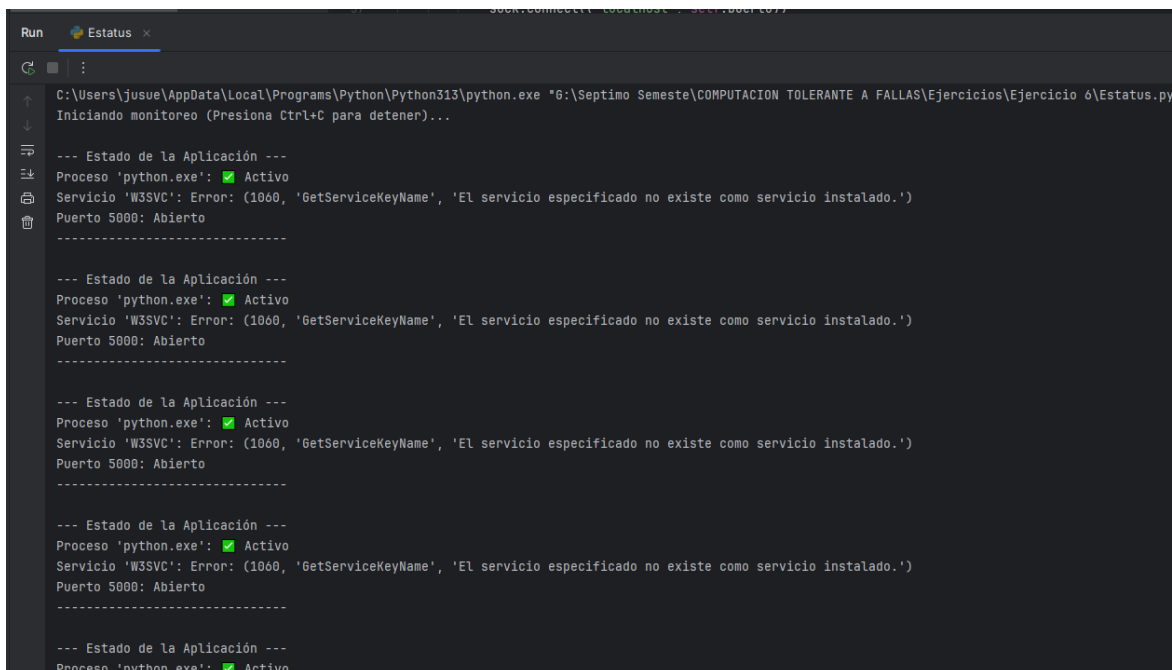
--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [ ] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

```



```
Run Estatus x
C:\Users\jusue\AppData\Local\Programs\Python\Python313\python.exe "G:\Septimo Semestre\COMPUTACION TOLERANTE A FALLAS\Ejercicios\Ejercicio 6\Estatus.py"
Iniciando monitoreo (Presiona Ctrl+C para detener)...

--- Estado de la Aplicación ---
Proceso 'python.exe': [✓] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [✓] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [✓] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [✓] Activo
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')
Puerto 5000: Abierto
-----

--- Estado de la Aplicación ---
Proceso 'python.exe': [✓] Activo
```

Servicio Inactivo:



[illegible]

```
C:\Users\josue\AppData\Local\Programs\Python\Python313\python.exe "G:\Septimo Semestre\COMPUTACION TOLERANTE A FALLAS\Ejercicios\Ejercicio 6\Estatus.py"  
Iniciando monitoreo (Presiona Ctrl+C para detener)...
```

```
--- Estado de la Aplicación ---  
Proceso 'python.exe': ✔ Activo  
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')  
Puerto 5000: Cerrado  
-----
```

```
--- Estado de la Aplicación ---  
Proceso 'python.exe': ✔ Activo  
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')  
Puerto 5000: Cerrado  
-----
```

```
--- Estado de la Aplicación ---  
Proceso 'python.exe': ✔ Activo  
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')  
Puerto 5000: Cerrado  
-----
```

```
--- Estado de la Aplicación ---  
Proceso 'python.exe': ✔ Activo  
Servicio 'W3SVC': Error: (1060, 'GetServiceKeyName', 'El servicio especificado no existe como servicio instalado.')  
Puerto 5000: Cerrado  
-----
```