

## TABLA DE CONTENIDOS

1. Descripción del Sistema
  2. Requisitos Técnicos
  3. Instalación Rápida
  4. Primeros Pasos
  5. Uso del Sistema
  6. Ejemplos Prácticos
  7. Solución de Problemas
  8. Referencia Técnica
- 

## 1. DESCRIPCIÓN DEL SISTEMA

### 1.1 ¿Qué es el Sistema de Monitoreo de Accesos?

El **Sistema de Monitoreo de Accesos** es una aplicación desarrollada en Python diseñada para:

-  **Monitorear** accesos a servidores y sistemas
-  **Detectar** actividades sospechosas automáticamente
-  **Generar** reportes detallados de seguridad
-  **Exportar** datos para análisis posterior

### 1.2 Características Principales

Característica	Descripción	Beneficio
<b>Registro en Tiempo Real</b>	Captura cada intento de acceso	Visibilidad completa
<b>Detección Automática</b>	Identifica patrones sospechosos	Prevención proactiva
<b>Reportes Detallados</b>	Estadísticas y análisis	Toma de decisiones informadas
<b>Múltiples Formatos</b>	CSV y JSON export	Flexibilidad de uso

---

## 2. REQUISITOS TÉCNICOS

### 2.1 Requisitos Mínimos del Sistema

Componente	Requerimiento
<b>Sistema Operativo</b>	Windows 10+, Linux, macOS
<b>Python</b>	Versión 3.8 o superior
<b>Memoria RAM</b>	512 MB mínimo
<b>Espacio Disco</b>	100 MB disponibles

## 2.2 Bibliotecas Requeridas

```
python
# Bibliotecas incluidas (no requieren instalación)

import random
import datetime
from collections import defaultdict
import csv
import json

# Bibliotecas opcionales (para funcionalidades extendidas)
# pip install pandas matplotlib seaborn
```

## 3. INSTALACIÓN RÁPIDA

### 3.1 Pasos de Instalación

#### Paso 1: Verificar Python

```
bash
python --version
# Debe mostrar: Python 3.8 o superior
```

#### Paso 2: Crear Directorio del Proyecto

```
bash
mkdir sistema-monitoreo
cd sistema-monitoreo
```

#### Paso 3: Guardar el Código

- Crear archivo: sistema\_monitoreo.py
- Copiar el código proporcionado

#### Paso 4: Ejecutar por Primera Vez

```
bash  
python sistema_monitoreo.py
```

### 3.2 Estructura de Archivos

```
text  
sistema-monitoreo/  
|   └── sistema_monitoreo.py    # Código principal  
|   └── reporte_accesos_*.csv  # Reportes en CSV  
|   └── reporte_accesos_*.json # Reportes en JSON  
└── manual_usuario.pdf      # Este manual
```

---

## 4. PRIMEROS PASOS

### 4.1 Inicialización del Sistema

```
python  
# Importar y crear el sistema  
from sistema_monitoreo import SistemaMonitoreoAccesos
```

```
sistema = SistemaMonitoreoAccesos()
```

### 4.2 Configuración Inicial Automática

El sistema incluye configuración predefinida:

- **6 usuarios de ejemplo:** admin, usuario1, usuario2, invitado, root, test\_user
  - **5 servidores:** 192.168.1.10, 192.168.1.20, 192.168.1.30, 10.0.0.5, servidor\_bd
  - **Umbral de alertas:** 3 intentos fallidos
  - **Horas sospechosas:** 00:00 - 05:59
- 

## 5. USO DEL SISTEMA

### 5.1 Registro de Intentos de Acceso

#### Ejemplo: Acceso Exitoso

```
python
```

```
sistema.registrar_intento(  
    usuario="admin",  
    servidor="192.168.1.10",  
    ip="192.168.1.100",  
    tipo_acceso=1 # 1 = ÉXITO  
)
```

#### **Ejemplo: Acceso Fallido**

```
python  
sistema.registrar_intento(  
    usuario="usuario1",  
    servidor="10.0.0.5",  
    ip="192.168.1.200",  
    tipo_acceso=0 # 0 = FALLIDO  
)
```

#### **5.2 Simulación Automática**

```
python  
# Generar 25 intentos de acceso automáticamente  
sistema.simular_accesos(25)
```

#### **5.3 Generación de Reportes**

##### **Reporte Completo**

```
python  
sistema.mostrar_reporte()
```

##### **Reporte por Usuario Específico**

```
python  
sistema.mostrar_reporte(usuario="admin")
```

#### **5.4 Sistema de Alertas**

```
python  
# Obtener y mostrar alertas  
alertas = sistema.generar_alertas()
```

```
for numero, alerta in enumerate(alertas, 1):
```

```
print(f"{{numero}. {alerta}}")
```

## 5.5 Exportación de Datos

```
python
```

```
# Exportar a CSV (Excel)
```

```
sistema.exportar_reportes('csv')
```

---

```
# Exportar a JSON
```

```
sistema.exportar_reportes('json')
```

---

## 6. EJEMPLOS PRÁCTICOS

### 6.1 Ejemplo 1: Monitoreo Diario

```
python
```

```
# Configuración para monitoreo diario
```

```
sistema = SistemaMonitoreoAccesos()
```

```
# Simular actividad del día
```

```
sistema.simular_accesos(50)
```

```
# Revisar seguridad
```

```
alertas = sistema.generar_alertas()
```

```
# Guardar evidencias
```

```
sistema.exportar_reportes('csv')
```

```
print(f"Monitoreo diario completado. {len(alertas)} alertas encontradas.")
```

### 6.2 Ejemplo 2: Investigación Específica

```
python
```

```
# Enfocarse en usuario sospechoso
```

```
sistema.mostrar_reporte(usuario="invitado")
```

```
# Filtrar alertas del usuario
```

```
alertas_especificas = [a for a in sistema.generar_alertas()  
    if "invitado" in a.lower()]
```

```
print("🔍 Alertas específicas del usuario:")
```

```
for alerta in alertas_especificas:
```

```
    print(f" • {alerta}")
```

### 6.3 Ejemplo 3: Personalización Avanzada

```
python
```

```
# Configurar parámetros de seguridad
```

```
sistema.umbral_intentos_fallidos = 5
```

```
sistema.horas_sospechosas = [22, 23, 0, 1, 2, 3, 4] # 10PM-4AM
```

---

```
# Agregar elementos personalizados
```

```
sistema.usuarios.append("nuevo_empleado")
```

```
sistema.servidores.append("192.168.2.100")
```

```
sistema.ips_sospechosas.append("10.1.1.200")
```

---

## 7. SOLUCIÓN DE PROBLEMAS

### 7.1 Problemas Comunes y Soluciones

Problema	Causa Probable	Solución
"python no se reconoce"	Python no en PATH	Usar python3 o reinstalar Python
"Módulo no encontrado"	Bibliotecas faltantes	Ejecutar pip install pandas
"Permiso denegado"	Sin permisos escritura	Cambiar directorio o ejecutar como administrador
Reportes vacíos	Sin datos registrados	Usar simular_accesos(20) primero
Sin alertas	Umbral muy alto	Reducir umbral_intentos_fallidos

### 7.2 Códigos de Error

Código	Significado	Acción
<b>ERR PYTHON</b>	Python no instalado	Descargar de <a href="https://www.python.org">python.org</a>
<b>ERR IMPORT</b>	Error importación	Verificar bibliotecas
<b>ERR FILE</b>	Error archivo	Verificar permisos
<b>ERR DATA</b>	Datos corruptos	Reiniciar sistema