

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Sistema para la Gestión y Administración de la Configuración de Software**

# Curso: Calidad y Pruebas de Software

# Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

| **Sebastian Arce Bracamonte** | **(2019062886)** |
| --- | --- |
| **Chata Choque, Brant Antony** | **(2020067577)** |
| **Renzo Fernando Loyola Vilca Choque** | **(2021072615)** |

**Tacna – Perú**

**2025**

**Desarrollo de un sistema híbrido en C++ y Python basado en Machine Learning para la detección y supresión automática de keyloggers en entornos Windows**

**Documento de Visión**

**Versión 1.0**

| **CONTROL DE VERSIONES** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | SAB | BCC | FRLV | 16/8/2025 | Primera Versión |

**ÍNDICE GENERAL**

[**Curso: Calidad y Pruebas de Software**](#) **1**

[**Docente: Mag. Patrick Cuadros Quiroga**](#) **1**

[1. Introducción](#_rhxrdtdcvsvm) 5

[1.1 Propósito](#_612qz8x8pf6w) 5

[1.2 Alcance](#_37331ohaxgg) 5

[1.3 Definiciones, Siglas y Abreviaturas](#_y3823smosrek) 6

[1.4 Referencias](#_l2sqm2t5m892) 6

[2. Visión General](#_4d1ssoecytzv) 6

[2.1 Posicionamiento](#_12d5uuuy3w78) 6

[2.2 Oportunidad de Negocio](#_mxtqecl63vz3) 7

[2.3 Definición del Problema](#_nuy49bs5gwp2) 7

[3. Descripción de los Interesados y Usuarios](#_6lcixj92y66x) 7

[3.1 Resumen de los Interesados](#_vsl7gyy4d54e) 7

[3.2 Resumen de los Usuarios](#_9crxr3xgxj27) 7

[3.3 Entorno del Usuario](#_btwqx4djxj7a) 7

[4. Vista General del Proyecto](#_atq2tkqcyxnv) 8

[4.1 Perspectiva del Producto](#_wxiohwfbfmmf) 8

[4.2 Resumen de Capacidades](#_jxhm1ufj8mp) 8

[4.3 Suposiciones y Dependencias](#_8t0qwa344ymr) 8

[5. Características del Proyecto](#_oq7g6tw1cs99) 8

[6. Restricciones](#_o2f88ri1dply) 9

[7. Satisfacción al Cliente](#_481rh7cmceoa) 9

10

## **1. Introducción**

### **1.1 Propósito**

Este documento articula la visión integral del sistema híbrido en C++ y Python con Machine Learning para la detección y supresión automática de keyloggers en entornos Windows. Se busca alinear los objetivos técnicos y de seguridad con las necesidades de los usuarios, asegurando una solución robusta, innovadora y escalable en ciberseguridad preventiva.

### **1.2 Alcance**

**El sistema se enfocará en:**

* Monitoreo de procesos, tráfico de red y registros de teclado.
* Identificación de comportamientos sospechosos en tiempo real.
* Clasificación automática de procesos mediante modelos de ML.
* Supresión inmediata de keyloggers detectados y cuarentena de sus archivos.
* Validación experimental en entornos virtualizados seguros (VirtualBox + Cuckoo Sandbox).

### **1.3 Definiciones, Siglas y Abreviaturas**

* **ML:** Machine Learning.
* **C++:** Lenguaje de programación de bajo nivel usado para módulos de monitoreo.
* **Python:** Lenguaje de programación de alto nivel para la implementación de modelos de ML.
* **Keylogger:** Software malicioso diseñado para registrar pulsaciones de teclado.
* **API Hook:** Técnica para interceptar funciones del sistema.

### **1.4 Referencias**

* Bishop, M. Computer Security: Art and Science.
* Papers IEEE y Scopus sobre detección de malware con ML.
* Documentación de Pybind11 y Named Pipes en Windows.

## **2. Visión General**

### **2.1 Posicionamiento**

El sistema se presenta como una herramienta avanzada de ciberseguridad, orientada a detectar keyloggers incluso en sus versiones polimórficas, superando las limitaciones de los antivirus tradicionales basados en firmas.

### **2.2 Oportunidad de Negocio**

Con el incremento de ataques de robo de credenciales y espionaje corporativo, este sistema representa un valor estratégico tanto para usuarios finales como para empresas que requieran seguridad proactiva contra amenazas invisibles.

### **2.3 Definición del Problema**

Los keyloggers actúan de manera oculta en Windows, disfrazándose de procesos legítimos y evadiendo sistemas de protección tradicionales. El problema radica en la falta de soluciones que detecten comportamientos anómalos en tiempo real, lo que expone información crítica como contraseñas y datos financieros.

## **3. Descripción de los Interesados y Usuarios**

### **3.1 Resumen de los Interesados**

* Docente y asesores académicos: Supervisan la viabilidad y cumplimiento de objetivos.
* Empresas/Usuarios: Se beneficiarán del uso del sistema para reforzar su seguridad digital.

### **3.2 Resumen de los Usuarios**

* Administrador de Seguridad: Responsable de la configuración del sistema y validación de resultados.
* Usuarios Finales: Obtendrán protección automática sin necesidad de conocimientos técnicos avanzados.

### **3.3 Entorno del Usuario**

El sistema se usará en Windows 10/11, con requisitos mínimos de CPU Intel i5, 8 GB RAM y 200 MB de espacio libre.

## **4. Vista General del Proyecto**

### **4.1 Perspectiva del Producto**

Un sistema híbrido que combina:

* C++ para bajo nivel (captura de procesos, hooks, tráfico).
* Python para análisis y ML.
* Comunicación en tiempo real mediante Named Pipes o Pybind11.

### **4.2 Resumen de Capacidades**

| **Capacidad** | **Beneficio** |
| --- | --- |
| **Detección en tiempo real** | **Previene robo de datos sensibles.** |
| Clasificación ML | Diferencia procesos legítimos vs. maliciosos. |
| Supresión automática | Finaliza procesos sospechosos y mueve archivos a cuarentena. |
| Informes y Alertas | Genera reportes para administradores de seguridad. |
| Integración híbrida | Aprovecha la velocidad de C++ y la flexibilidad de Python. |

### **4.3 Suposiciones y Dependencias**

* El sistema se ejecuta en Windows con permisos de administrador.
* Se requiere conexión a internet para entrenar/actualizar modelos.
* El desempeño depende de librerías externas como scikit-learn, XGBoost, Pybind11.

## **5. Características del Proyecto**

* **Usabilidad:** Interfaz sencilla para monitorear procesos y reportes.
* **Consistencia:** Estructura uniforme en reportes y notificaciones.
* **Disponibilidad:** Operación continua en segundo plano.
* **Seguridad:** Cifrado en logs y comunicaciones.
* **Mantenibilidad:** Código modular en C++/Python.
* **Inmediatez:** Respuesta de detección en menos de 2 segundos.

## **6. Restricciones**

* **Tiempo de desarrollo:** 6 meses**.**
* Equipo de máximo 3 integrantes.
* Presupuesto limitado a software libre y hardware existente.

## **7. Satisfacción al Cliente**

Medida por precisión del modelo (>95%), baja tasa de falsos positivos (<5%) y supresión exitosa de keyloggers.