Tipo de artículo: Artículo original

Análisis de ficheros de registros de accesos de servidores web Nginx y Apache

Analysis of files of access logs from Nginx and Apache web servers

Dariel González Robinson[1*], https://orcid.org/0009-0007-8105-6880
Yohandra Echavarria Castillo[1], https://orcid.org/0000-0001-6163-2819
Madelín Haro Pérez[1], https://orcid.org/0000-0002-5700-8244
Mónica Peña Casanova [1], https://orcid.org/0000-0003-2500-4510

[1] Universidad de las Ciencias Informáticas. Cuba.

*Autor para la correspondencia. dgonzalezr@estudiantes.uci.cu

RESUMEN

Este artículo se centra en el análisis de registros de acceso para obtener información valiosa sobre el tráfico de un sitio web, el comportamiento del usuario y posibles problemas de seguridad. Se desarrolló un script para limpiar y analizar los datos generados por sistemas web que utilizan Nginx o Apache. El script, escrito en Bash, permite la limpieza, análisis y visualización de trazas, identificando eventos significativos. El script puede analizar los registros de acceso de ambos servidores web, así como de otras tecnologías web que utilicen el mismo formato estándar para el almacenamiento de las trazas. Para la experimentación se utilizó el script sobre un conjunto de trazas generados por el sistema de gestión XABAL EXCRIBA. Los resultados obtenidos a partir del análisis de las trazas determinan que el sitio web EXCRIBA se utiliza principalmente para la consulta de información debido a la gran cantidad de peticiones GET. Además, los

resultados muestran que el servicio es confiable y eficiente en la entrega de contenido o servicios debido a la cantidad de respuestas con código exitoso. Sin embargo, la presencia de redirecciones y errores sugiere que hay áreas de mejora en la optimización de rutas o enlaces y en la forma en que las solicitudes son manejadas por el servidor.

Palabras clave: análisis de datos; análisis de trazas; script; bash; Nginx; Apache.

ABSTRACT

This article focuses on analyzing access logs to obtain valuable information about a website's traffic, user behavior, and potential security issues. A script was developed to clean and analyze data generated by web systems using Nginx or Apache. The script, written in Bash, allows cleaning, analysis and visualization of traces, identifying significant events. The script can analyze access logs from both web servers, as well as other web technologies that use the same standard format for storing traces. For the experimentation, the script was used on a set of traces generated by the XABAL EXCRIBA management system. The results obtained from the analysis of the traces determine that the EXCRIBA website is mainly used for information consultation due to the large number of GET requests. Furthermore, the results show that the service is reliable and efficient in delivering content or services due to the number of responses with successful code. However, the presence of redirects and errors suggests that there are areas for improvement in route or link optimization and in the way requests are handled by the server.

Keywords: analysis of data; trace analysis; script; bash; Nginx; Apache.

Recibido: 15/10/2024
Aceptado: 09/12/2024
En línea: 01/01/2025

Introducción

Hoy en día con los avances tecnológicos en especial los que tienen que ver directamente con el manejo y procesamiento de la información, han facilitado de forma significativa la labor de las organizaciones en

general. Es por ello que ante el crecimiento y desarrollo de las tecnologías de información también son más las amenazas a las que una organización debe hacer frente (Ghiasi et al., 2021).

Actualmente se podría afirmar que la mayoría de los sistemas y medios tecnológicos generan trazas ante cualquier evento que se esté suscitando en el sistema, aplicación o estructura tecnológica (Landauer et al., 2020). La gestión de trazas aporta un valor agregado a la seguridad de la información dentro de las organizaciones. Esta información analizada y gestionada adecuadamente podría convertirse en una base de datos de incidentes y eventos con utilidad en diversos fines entre los cuales se encuentran: la administración de recursos, detección de intrusiones, la resolución de problemas, análisis forense y auditorías (Le & Zhang, 2021).

Gran parte de los sistemas web garantizan la generación de trazas y el almacenamiento de estos en ficheros pero no realizan tareas de procesamiento y análisis de los datos (Zhu et al., 2023), por lo que se recomienda implementar una solución de análisis de trazas que pueda procesar y analizar los datos para identificar tendencias.

Ante la problemática anteriormente expuesta se desarrolló un script con el objetivo de limpiar y analizar los datos generados por sistemas web que utilizan Nginx o Apache como tecnologías para sus servidores web, así como de otros servidores web que almacenen las trazas en el mismo formato estándar de estas tecnologías.

Métodos o Metodología Computacional

Para el desarrollo de esta investigación se emplearon los siguientes métodos científicos:

- Analítico-Sintético: Se utiliza para descomponer los registros en partes más pequeñas, como identificar solicitudes GET, códigos de respuesta y errores. Luego, se sintetiza esta información para obtener patrones generales del tráfico web y áreas de mejora en el sistema.
- Inductivo-Deductivo: A partir de los datos concretos de los registros, se generan conclusiones generales, como la prevalencia de ciertos tipos de solicitudes. A partir de estas generalizaciones, se formulan hipótesis sobre el comportamiento del tráfico o posibles mejoras en la configuración del servidor.

- Experimental: Se aplica durante la ejecución del script sobre los datos reales del sistema XABAL EXCRIBA. Este método permitió observar cómo las diferentes configuraciones del script afectan los resultados obtenidos.

Diseño del Script

El script fue diseñado con el objetivo de proporcionar una herramienta para la limpieza y análisis de logs en sistemas Unix. Bash fue seleccionado como el lenguaje de scripting debido a su amplia disponibilidad en la mayoría de los sistemas operativos basados en Unix y Linux, así como por su reconocida eficiencia en la ejecución de tareas relacionadas con el procesamiento de texto y archivos. La utilización de Bash permite una integración directa con el shell del sistema y la ejecución de comandos de manera secuencial o condicional, lo que es esencial para el manejo de flujos de trabajo complejos en la limpieza y análisis de datos.

Análisis de Logs

El análisis de logs se centró en identificar eventos significativos y tendencias. Se utilizaron técnicas como el conteo de ocurrencias (Le & Zhang, 2021) y la agrupación de entradas por categorías (He et al., 2021). Además, se implementaron métodos para resumir los datos, como la generación de estadísticas descriptivas y la visualización de los resultados en forma de tablas o gráficos, lo que facilita la interpretación y la toma de decisiones basada en los datos analizados.

Fundamentación de la herramienta y el lenguaje de desarrollo

La elección de Bash y las herramientas de línea de comandos asociadas se justifica por la necesidad de un sistema que pueda ejecutarse con recursos limitados y que sea altamente personalizable. Las herramientas como *grep*, *awk* y *sed* son estándares de facto para el procesamiento de texto en entornos Unix y proporcionan una manera rápida y eficiente de manipular grandes volúmenes de datos sin la necesidad de interfaces gráficas o software adicional.

Reproducibilidad

Para asegurar que otros investigadores puedan replicar el estudio, se incluyó una documentación del script[1], describiendo cada parámetro y su función en el proceso. Esto permitirá un uso eficiente del script a la hora

[1] Disponible en https//github.com/dgrobinson0/analisis_de_logs

de realizar un análisis posterior sobre estos datos. Además, se incluyeron comentarios detallados en el código permitiendo que partes del mismo puedan ser reutilizadas o modificadas para adaptarse a diferentes conjuntos de datos o requisitos de análisis.

Resultados y discusión

Nginx y Apache son dos de los servidores web más populares en la actualidad para servir contenido en Internet. Ambos generan trazas y las almacenan en ficheros de dos tipos:

1. Access logs (registros de accesos): Registran información sobre cada solicitud recibida.
2. Error logs (registros de errores): Registran cualquier error o advertencia encontrada.

El script desarrollado se centra en el análisis de las trazas almacenadas en los registros de accesos. Dado que Nginx y Apache siguen los mismos estándares para el almacenamiento de las trazas, el script puede analizar los registros de acceso de ambos servidores web así como de otras tecnologías web que utilicen el mismo formato estándar para el almacenamiento de las trazas.

El script cuenta con varias opciones de filtrado de datos para facilitar la visualización de la información, permite el filtrado por una dirección IP, fecha de la solicitud, método utilizado en la solicitud y/o código de respuesta de estado de la petición. Además, permite generar un reporte sobre las direcciones IP, fechas más activas y recursos más solicitados, así como los horarios en que se realiza una mayor cantidad de solicitudes. En la Figura 1 se muestra el panel de ayuda de la herramienta donde se visualizan los parámetros que admite el script.

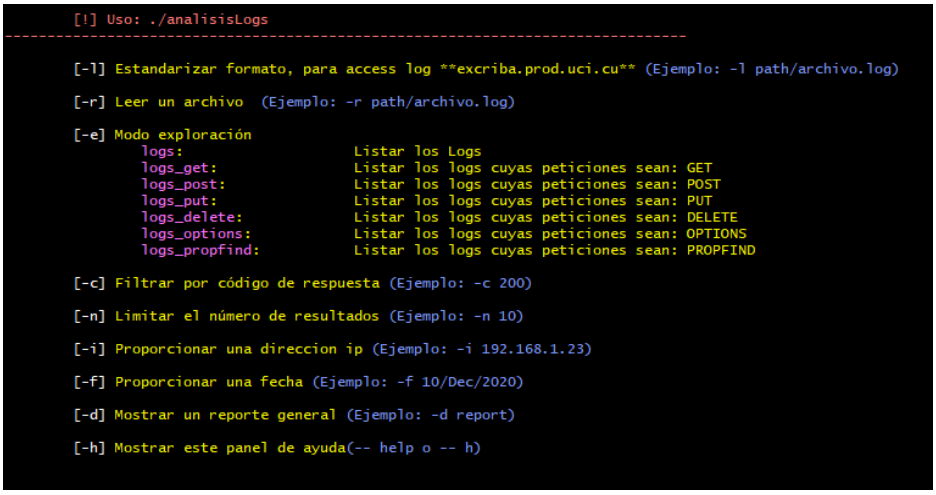

Fig 1: Panel de ayuda del script.

Para la experimentación se utilizó el script sobre un conjunto de logs generados por el sistema de gestión XABAL EXCRIBA[2] que propicia la generación de documentos, su revisión, administración, distribución, custodia y disposición. Utiliza Apache Tomcat como servidor web y cuenta con la capacidad para generar registros de actividad o trazas basados en el acceso de los usuarios. Los datos que conforman estas trazas generadas por el sistema se clasifican en diferentes tipos basados en sus propias características y usos (Kantardzic, 2011):

1. Datos numéricos: Son aquellos que se expresan en números y pueden ser variables o enteros. Estos datos son cuantificables y permiten realizar operaciones matemáticas.
2. Datos categóricos: Son aquellos que representan características y se agrupan en categorías. Estos datos no son cuantificables y suelen ser el resultado de clasificar o describir atributos de una población.

En la Tabla 1 se muestran los campos de las trazas que se utilizarán en el proceso de análisis, así como la descripción y su clasificación atendiendo al tipo de dato según sus características.

Tabla 1: Descripción y clasificación de los campos a usar en el análisis.

Campos	Tipo de datos	Descripción
Dirección IP	Categórica	Identifica al cliente que realiza la solicitud

[2] https://excriba.uci.cu/page/

Fecha/ Hora/ Zona Horaria	Categórica	Registra cuándo se hizo la solicitud
Método	Categórica	Contiene el método HTTP de la petición realizada
URL	Categórica	Ubicación del recurso solicitado
Versión	Categórica	Versión del protocolo HTTP utilizado
Código de Estado	Numérico	Código de respuesta del servidor

Proceso de análisis de datos (Etapa: Limpieza de los datos)

El proceso de limpieza y normalización de datos es un paso fundamental en la preparación para un análisis de datos eficiente y confiable. La normalización de datos es el proceso sistemático de la descomposición de los datos para eliminar la redundancia en la información y las características no deseadas que pueden ser generadas en el momento de insertar, actualizar y eliminar registros (Amiri et al., 2024)

Comenzando con una colección de archivos de acceso al sistema XABAL EXCRIBA, el objetivo fue centralizar esta información. Se contaba con 98 ficheros que contenían información de acceso al sistema desde el 24 de noviembre del 2023 hasta el 2 de marzo del 2024. Se transfirieron todos estos datos a un archivo unificado, creando así una fuente de datos cohesiva. Una vez consolidados los datos, se eliminan las redundancias en los datos. Los datos repetidos no solo inflan artificialmente el tamaño del conjunto de datos, sino que también conducen a sesgos en el análisis.

Para garantizar el cumplimiento de esta etapa se utilizó la función “estandar_archivo_excriba” propia del script creado la cual hace posible eliminar campos con información repetida de manera innecesaria obteniendo los datos limpios, listos para ser analizados. En la Figura 2 se muestran las trazas generadas por defecto por el sistema, mientras que en la Figura 3 se muestran las trazas luego del proceso de limpieza de los datos.

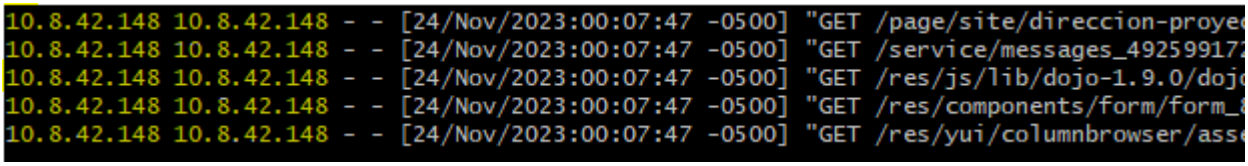

Fig 2: Estructura de las trazas antes del proceso de limpieza de los datos.

```
10.8.42.148  - - [24/Nov/2023:00:07:47 -0500] "GET /page/site/direccion-proyectos-e
10.8.42.148  - - [24/Nov/2023:00:07:47 -0500] "GET /service/messages_49259917243123
10.8.42.148  - - [24/Nov/2023:00:07:47 -0500] "GET /res/js/lib/dojo-1.9.0/dojo/dojo
10.8.42.148  - - [24/Nov/2023:00:07:47 -0500] "GET /res/components/form/form_865ad6
10.8.42.148  - - [24/Nov/2023:00:07:47 -0500] "GET /res/yui/columnbrowser/assets/sk
```

Fig 3: Estructura de las trazas luego del proceso de limpieza de los datos.

Nótese que posteriormente al proceso de limpieza y normalización de los datos los dos primeros campos de cada traza pasan a conformar un solo campo.

Proceso de análisis de datos (Etapa: Análisis de los datos)

En esta sección se analizan los resultados que fueron obtenidos luego de varias ejecuciones del script desarrollado alternando el uso de los parámetros admitidos. En la Figura 4 se presenta la cantidad de solicitudes realizadas distribuidas por los métodos que se utilizaron para un total de 994637.

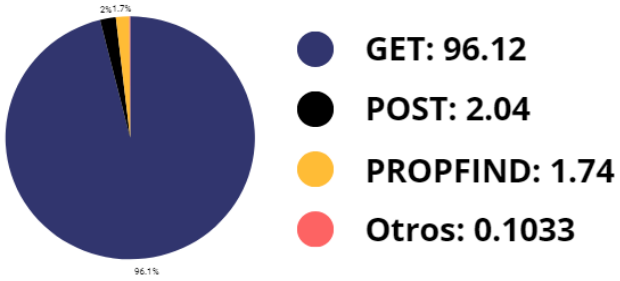

Fig 4: Cantidad de solicitudes distribuidas por métodos.

A partir de los resultados obtenidos anteriormente se plantea que el servicio web en cuestión está predominantemente orientado a la consulta de información dado que:

1. La gran cantidad de peticiones GET en comparación con otros métodos indica que los usuarios acceden al servicio principalmente para leer o descargar datos.
2. La baja frecuencia de métodos como POST, PUT, y DELETE sugiere que las interacciones que implican cambios en el estado del servidor como la creación, actualización o eliminación de recursos son mucho menos comunes.

Como parte del análisis también se identifican las fechas claves donde hubo un mayor tráfico de solicitudes y las direcciones IP que estuvieron más activas. El conocimiento de cuándo se producen picos de tráfico

permite a los administradores de sistemas planificar y escalar recursos adecuadamente para manejar las cargas de trabajo. Por otra parte, identificar las direcciones IP más activas puede ayudar a detectar posibles ataques o actividades sospechosas. En la Figura 5 se muestran las fechas claves determinadas por los altos picos de tráficos ocasionados.

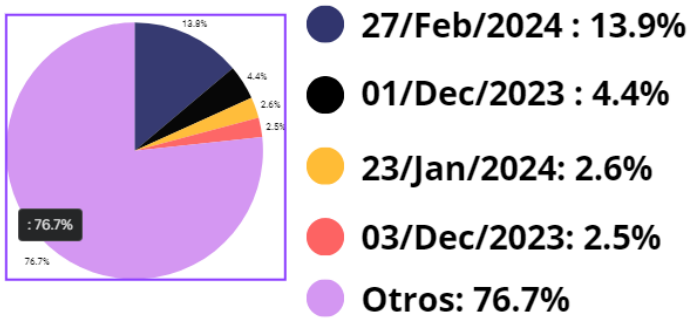

Fig 5: Fechas con mayor índice de tráfico.

Se debe destacar en el resultado anterior que los días seleccionados como fechas con mayor índice de tráfico, representan casi ¼ del total de tráfico generado en los 98 días analizados. Además, sería interesante investigar por qué el 27 de febrero de 2024 tiene un porcentaje tan alto en comparación con las otras fechas. Para profundizar en el análisis, en la Tabla 2 se muestran los horarios y direcciones IP donde se realizaron mayores solicitudes al sistema.

Tabla 2: Horarios y direcciones IP con mayor tráfico.

Fecha	Solicitudes	00:00 – 06:00	07:00 – 19:00	20:00 – 23:00	Dirección IP
27/Feb/2024	137785	0%	98.56%	1.44	10.58.12.16
01/Dec/2023	43348	2.51	45.64	51.85%	10.8.130.54
23/Jan/2024	26193	4%	95.83%	0.17	10.58.3.251
03/Dec/2023	24237	13.94%	50.51%	35.55	10.8.40.20

En el análisis de los datos, es fundamental entender los códigos de respuesta HTTP, ya que proporcionan información crucial sobre el estado de las solicitudes realizadas al servidor. En la Figura 6 se muestran los códigos de respuesta de estado por porcentaje. A continuación, se describen los códigos de respuesta HTTP más comunes:

1. **Códigos 200:** Indica que la solicitud ha tenido éxito.

2. **Códigos 300:** Códigos de redirección que indica que la solicitud tiene más de una posible respuesta.
3. **Códigos 400:** Indica que el servidor no puede o no procesará la solicitud debido a algo que se percibe como un error del cliente.
4. **Códigos 500:** Indica que el servidor encontró una condición inesperada que le impidió cumplir con la solicitud.

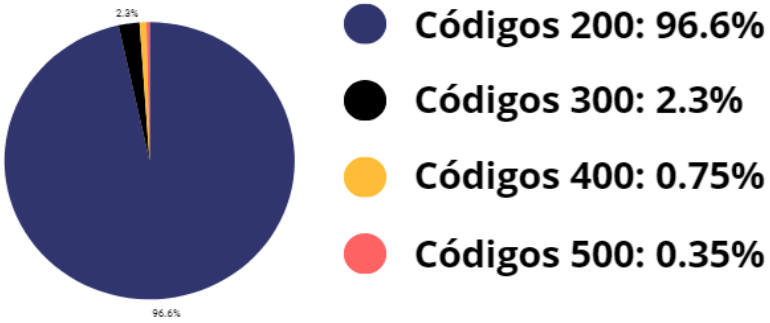

Fig 6: Códigos de respuesta de estados.

El análisis de los códigos de respuesta HTTP revela una visión general del estado de las solicitudes procesadas por el servidor EXCRIBA. Con un 96.6% de respuestas con código exitoso, indica un servicio confiable y eficiente en la entrega de contenido o servicios. Sin embargo, no todo es perfecto, ya que se observa una cantidad moderada de redirecciones lo cual podría sugerir la necesidad de optimizar ciertas rutas o enlaces para evitar pasos adicionales en el acceso a la información.

Por otro lado, los códigos de error de cliente y error de servidor reflejan áreas de posible mejora. Con 7,410 errores del cliente y 3,497 errores del servidor, es evidente que existen problemas tanto en la forma en que las solicitudes son realizadas por los clientes como en la capacidad del servidor para manejarlas adecuadamente. Estos errores no solo afectan la experiencia del usuario, sino que también pueden ser indicativos de problemas más profundos en la infraestructura o en la lógica de la aplicación.

Conclusiones

El análisis realizado en este trabajo ha permitido interpretar los resultados obtenidos a través del script preexistente aplicado sobre los registros de acceso del sistema de gestión XABAL EXCRIBA. Este análisis

contribuye significativamente al campo del análisis de la información de la empresa o entidad, ya que permite entender y gestionar adecuadamente la información contenida en los logs, convirtiéndola en una base de datos de incidentes y eventos útil para diversos fines.

Los resultados obtenidos a partir del análisis de los logs han proporcionado información valiosa sobre el uso del sistema XABAL EXCRIBA concluyendo que el servicio está orientado a la consulta de información ya que la mayoría de las solicitudes al sistema son para leer o descargar datos. También ha permitido identificar áreas de mejora para el sistema XABAL EXCRIBA observando que algunas solicitudes al sistema que resultan en errores podrían optimizarse para mejorar la experiencia del usuario.

Referencias

Amiri, Z., Heidari, A., Navimipour, N. J., Unal, M., & Mousavi, A. (2024). Adventures in data analysis: A systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems. *Multimedia Tools and Applications*, *83*(8), 22909-22973. https://link.springer.com/article/10.1007/s11042-023-16382-x

Ghiasi, M., Dehghani, M., Niknam, T., Kavousi-Fard, A., Siano, P., & Alhelou, H. H. (2021). Cyber-attack detection and cyber-security enhancement in smart DC-microgrid based on blockchain technology and Hilbert Huang transform. *IEEE Access*, *9*, 29429-29440. https://ieeexplore.ieee.org/abstract/document/9353530/

He, S., He, P., Chen, Z., Yang, T., Su, Y., & Lyu, M. R. (2021). A survey on automated log analysis for reliability engineering. *ACM computing surveys (CSUR)*, *54*(6), 1-37. https://dl.acm.org/doi/abs/10.1145/3460345

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons. https://books.google.es/books?hl=es&lr=&id=ZZ7l6v0CvRMC&oi=fnd&pg=PA1&dq=M.+Kantardzic+%2B+2019%2B+Data+Mining:+Concepts,+Models,+Methods,+and+Algorithms,+3+ed.,+John+Wiley+%26+Sons&ots=pQwjpjqDyg&sig=T9S6Ryrz2cMXsrffeax3M2xVvhg

Landauer, M., Skopik, F., Wurzenberger, M., & Rauber, A. (2020). System log clustering approaches for cyber security applications: A survey. *Computers & Security*, *92*, 101739. https://www.sciencedirect.com/science/article/pii/S0167404820300250

Le, V.-H., & Zhang, H. (2021). Log-based anomaly detection without log parsing. 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE),

Zhu, J., He, S., He, P., Liu, J., & Lyu, M. R. (2023). Loghub: A large collection of system log datasets for ai-driven log analytics. 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE),

Conflicto de interés

Los autores autorizan la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Dariel González Robinson, Yohandra Echavarria Castillo
2. Curación de datos: Dariel González Robinson, Yohandra Echavarria Castillo, Madelín Haro Pérez, Mónica Peña Casanova
3. Análisis formal: Dariel González Robinson, Yohandra Echavarria Castillo, Madelín Haro Pérez, Mónica Peña Casanova
4. Investigación: Dariel González Robinson, Yohandra Echavarria Castillo
5. Metodología: Madelín Haro Pérez
6. Administración del proyecto: Dariel González Robinson, Yohandra Echavarria Castillo
7. Recursos: Madelín Haro Pérez, Mónica Peña Casanova
8. Software: Dariel González Robinson, Yohandra Echavarria Castillo
9. Supervisión: Madelín Haro Pérez, Mónica Peña Casanova
10. Validación: Dariel González Robinson, Yohandra Echavarria Castillo
11. Visualización: Dariel González Robinson, Yohandra Echavarria Castillo

12. Redacción – borrador original: Dariel González Robinson, Yohandra Echavarria Castillo, Madelín Haro Pérez, Mónica Peña Casanova
13. Redacción – revisión y edición: Dariel González Robinson, Yohandra Echavarria Castillo, Madelín Haro Pérez, Mónica Peña Casanova

Financiación

La investigación no requirió fuente de financiamiento.