Evidencia GA5-220501106-AA3-EV01

Informe Técnico Protección de la Información

Aprendiz: Juan Carlos Lopez Moreno

Instructor: Alvaro Esteban Betancourt Matoma

Área Técnica

Servicio Nacional de Aprendizaje – SENA

Técnico en Programación de Aplicaciones y Servicios Para la Nube

Código: 3186263

Diciembre 2025

1. Introducción

El presente informe técnico documenta las estrategias de protección de la información, monitoreo y seguimiento implementadas en la plataforma web de comercio electrónico de Finca Miraflores. Esta plataforma, desarrollada con Next.js 14 y Supabase, constituye el canal principal de ventas y comunicación digital de la empresa, permitiendo la gestión de productos, procesamiento de transacciones y administración de usuarios en un entorno seguro y escalable.La arquitectura de la solución se fundamenta en un modelo monolítico modular optimizado para la nube, desplegado en Vercel con integración continua desde GitHub.

La base de datos relacional PostgreSQL gestionada por Supabase proporciona las capacidades de almacenamiento, autenticación y control de acceso necesarias para garantizar la confidencialidad, integridad y disponibilidad de la información empresarial y de los clientes.Este documento detalla las medidas de seguridad aplicadas en cada capa de la aplicación, desde la protección de rutas mediante middleware de autenticación hasta la validación de datos con esquemas tipados, pasando por el cifrado de comunicaciones y la gestión segura de tokens de sesión. Asimismo, se describen las herramientas y técnicas de monitoreo implementadas para supervisar el rendimiento, disponibilidad y comportamiento de la plataforma en tiempo real, permitiendo la detección temprana de anomalías y la respuesta ágil ante incidentes.

El enfoque de seguridad adoptado se alinea con las mejores prácticas de la industria, incorporando principios del framework NIST Cybersecurity Framework y controles específicos de OWASP para aplicaciones web. La implementación de Row Level Security en la base de datos, la validación exhaustiva de entradas mediante Zod, y la separación estricta entre credenciales públicas y privadas constituyen los pilares fundamentales de la estrategia de protección desplegada.

2. Alcance

Este informe abarca los aspectos de seguridad, monitoreo y operación de la plataforma web de Finca Miraflores en su versión 1.0, incluyendo todos los módulos funcionales especificados en el documento de requerimientos IEEE 830. El alcance técnico comprende la infraestructura de frontend desplegada en Vercel, la capa de servicios desarrollada en TypeScript, la base de datos PostgreSQL gestionada por Supabase, y las integraciones con servicios externos como Stripe para procesamiento de pagos y Resend para notificaciones por correo electrónico.

Las medidas de seguridad documentadas incluyen la protección de información sensible en todas las capas de la aplicación, específicamente en los módulos de autenticación y autorización, gestión de productos, procesamiento de órdenes de compra y administración de usuarios. Se contemplan tanto las rutas públicas accesibles a visitantes anónimos como las rutas protegidas reservadas para clientes autenticados y administradores, implementando controles de acceso diferenciados mediante roles.

El sistema de monitoreo cubre aspectos de rendimiento del frontend, disponibilidad de servicios backend, comportamiento de la base de datos, y métricas de uso de la aplicación. Se implementan herramientas nativas de las plataformas utilizadas, complementadas con soluciones de observabilidad que permiten el seguimiento de transacciones críticas y la detección de patrones anómalos en el tráfico de usuarios.

Quedan fuera del alcance de este informe los aspectos relacionados con la seguridad física del hardware de los usuarios finales, la gestión de redes corporativas de terceros, y los controles de seguridad internos de los proveedores de servicios en la nube utilizados. La

responsabilidad sobre estos aspectos recae en los respectivos proveedores de infraestructura conforme a sus acuerdos de nivel de servicio.

Informe Técnico de Ciberseguridad y Monitoreo

Sistema de Gestión y Venta de Café Orgánico - Finca Miraflores

1. Introducción

El presente informe técnico documenta las estrategias de protección de la información, monitoreo y seguimiento implementadas en la plataforma web de comercio electrónico de Finca Miraflores. Esta plataforma, desarrollada con Next.js 14 y Supabase, constituye el canal principal de ventas y comunicación digital de la empresa, permitiendo la gestión de productos, procesamiento de transacciones y administración de usuarios en un entorno seguro y escalable.

La arquitectura de la solución se fundamenta en un modelo monolítico modular optimizado para la nube, desplegado en Vercel con integración continua desde GitHub. La base de datos relacional PostgreSQL gestionada por Supabase proporciona las capacidades de almacenamiento, autenticación y control de acceso necesarias para garantizar la confidencialidad, integridad y disponibilidad de la información empresarial y de los clientes.

Este documento detalla las medidas de seguridad aplicadas en cada capa de la aplicación, desde la protección de rutas mediante middleware de autenticación hasta la validación de datos con esquemas tipados, pasando por el cifrado de comunicaciones y la gestión segura de tokens de sesión. Asimismo, se describen las herramientas y técnicas de monitoreo implementadas para supervisar el rendimiento, disponibilidad y comportamiento de la plataforma en tiempo real, permitiendo la detección temprana de anomalías y la respuesta ágil ante incidentes.

El enfoque de seguridad adoptado se alinea con las mejores prácticas de la industria, incorporando principios del framework NIST Cybersecurity Framework y controles específicos de OWASP para aplicaciones web. La implementación de Row Level Security en la base de datos, la validación exhaustiva de entradas mediante Zod, y la separación estricta entre credenciales públicas y privadas constituyen los pilares fundamentales de la estrategia de protección desplegada.

2. Alcance

Este informe abarca los aspectos de seguridad, monitoreo y operación de la plataforma web de Finca Miraflores en su versión 1.0, incluyendo todos los módulos funcionales especificados en el documento de requerimientos IEEE 830. El alcance técnico comprende la infraestructura de frontend desplegada en Vercel, la capa de servicios desarrollada en TypeScript, la base de datos PostgreSQL gestionada por Supabase, y las integraciones con servicios externos como Stripe para procesamiento de pagos y Resend para notificaciones por correo electrónico.

Las medidas de seguridad documentadas incluyen la protección de información sensible en todas las capas de la aplicación, específicamente en los módulos de autenticación y autorización, gestión de productos, procesamiento de órdenes de compra y administración de usuarios. Se contemplan tanto las rutas públicas accesibles a visitantes anónimos como las rutas protegidas reservadas para clientes autenticados y administradores, implementando controles de acceso diferenciados mediante roles.

El sistema de monitoreo cubre aspectos de rendimiento del frontend, disponibilidad de servicios backend, comportamiento de la base de datos, y métricas de uso de la aplicación. Se implementan herramientas nativas de las plataformas utilizadas, complementadas con soluciones

de observabilidad que permiten el seguimiento de transacciones críticas y la detección de patrones anómalos en el tráfico de usuarios.

Quedan fuera del alcance de este informe los aspectos relacionados con la seguridad física del hardware de los usuarios finales, la gestión de redes corporativas de terceros, y los controles de seguridad internos de los proveedores de servicios en la nube utilizados. La responsabilidad sobre estos aspectos recae en los respectivos proveedores de infraestructura conforme a sus acuerdos de nivel de servicio.

3. Protección de la Información

3.1 Confidencialidad

La protección de la confidencialidad de los datos se garantiza mediante múltiples capas de seguridad implementadas a lo largo de toda la arquitectura. El cifrado en tránsito se establece mediante el protocolo HTTPS con certificados TLS provistos automáticamente por Vercel, asegurando que toda comunicación entre el navegador del cliente y el servidor se transmita de forma cifrada. Este mecanismo protege contra ataques de interceptación y garantiza que información sensible como credenciales de acceso, datos de tarjetas de crédito y tokens de sesión no puedan ser capturados por terceros durante su transmisión.

El cifrado en reposo de la información almacenada en la base de datos PostgreSQL es gestionado por Supabase mediante cifrado AES-256, protegiendo los datos persistentes incluso en caso de acceso no autorizado al almacenamiento físico. Las contraseñas de usuarios nunca se almacenan en texto plano, sino que se procesan mediante el algoritmo bcrypt con un factor de trabajo de 10 iteraciones, generando hashes irreversibles que imposibilitan la recuperación de las credenciales originales en caso de filtración de la base de datos.

El control de acceso a la información se implementa mediante el sistema de autenticación JWT de Supabase, que genera tokens firmados digitalmente con una duración de sesión de una hora. Estos tokens contienen metadata del usuario incluyendo su rol, permitiendo que cada solicitud al servidor sea validada sin necesidad de consultar la base de datos repetidamente. El middleware de Next.js intercepta todas las peticiones a rutas protegidas y verifica la validez del token antes de permitir el acceso, denegando automáticamente solicitudes con tokens expirados o manipulados.

La separación de credenciales se gestiona rigurosamente mediante variables de entorno, distinguiendo entre claves públicas expuestas al navegador y claves secretas que permanecen exclusivamente en el servidor. Las claves de API de Stripe, el Service Role Key de Supabase y las credenciales de SMTP para envío de correos se almacenan únicamente en el entorno de ejecución serverless de Vercel, inaccesibles desde el código cliente. Esta segregación impide que usuarios malintencionados puedan extraer credenciales sensibles mediante inspección del código JavaScript del navegador.

3.2 Integridad

La integridad de los datos se preserva mediante validación exhaustiva de todas las entradas de usuario utilizando la librería Zod, que define esquemas tipados para cada formulario y endpoint de la aplicación. Antes de que cualquier dato ingresado por un usuario sea procesado o almacenado en la base de datos, es validado contra un esquema que especifica tipos de datos esperados, rangos permitidos, formatos requeridos y restricciones de longitud. Los datos que no cumplen con estas especificaciones son rechazados automáticamente, previniendo inyecciones SQL, cross-site scripting y otros ataques de manipulación de datos.

La auditoría de cambios en registros críticos se implementa mediante campos de timestamp en las tablas de la base de datos, registrando automáticamente la fecha y hora de creación y última modificación de cada registro. Las tablas de productos, órdenes y perfiles de usuario incluyen columnas created_at y updated_at que permiten rastrear el historial de modificaciones y detectar alteraciones no autorizadas. Si bien el sistema actual no incluye un log detallado de todas las operaciones, la estructura de la base de datos permite implementar triggers de PostgreSQL para capturar cambios específicos en tablas sensibles.

Las transacciones de compra se protegen mediante el uso de transacciones ACID en PostgreSQL, garantizando que operaciones complejas como la creación de una orden con múltiples productos se ejecuten de forma atómica. Si algún paso del proceso falla, como la actualización del inventario o el registro del pago, toda la transacción se revierte automáticamente, evitando inconsistencias en los datos que podrían derivar en sobreventa de productos o cobros sin orden registrada.

La validación de webhooks de Stripe constituye un control crítico para la integridad de las transacciones de pago. Cada notificación enviada por Stripe al endpoint de webhook incluye una firma digital generada con el secreto compartido entre Stripe y la aplicación. El servidor verifica esta firma antes de procesar la notificación, descartando automáticamente webhooks manipulados o generados por fuentes no autorizadas. Este mecanismo impide que atacantes puedan simular pagos exitosos enviando notificaciones falsas al sistema.

3.3 Disponibilidad

La alta disponibilidad de la plataforma se fundamenta en la arquitectura serverless de Vercel, que distribuye automáticamente la aplicación en múltiples nodos geográficos a través de una red de entrega de contenido global. Esta distribución garantiza que el frontend pueda servirse

desde el nodo más cercano al usuario, reduciendo latencia y proporcionando resiliencia ante fallos de infraestructura regional. La replicación automática del código en múltiples ubicaciones elimina puntos únicos de falla en la capa de presentación.

La base de datos PostgreSQL gestionada por Supabase opera con réplicas automáticas y respaldos continuos, garantizando recuperación ante fallos de hardware o corrupción de datos. Los backups se ejecutan cada 24 horas con retención de 7 días, permitiendo restaurar el sistema a cualquier punto en el tiempo durante la última semana. En caso de fallo catastrófico del datacenter primario, Supabase puede activar una réplica de lectura en menos de 15 minutos, minimizando el tiempo de inactividad.

El escalado automático de la infraestructura serverless permite que la plataforma responda dinámicamente a picos de tráfico sin intervención manual. Vercel incrementa automáticamente el número de instancias de función cuando el volumen de peticiones aumenta, manteniendo tiempos de respuesta consistentes incluso durante eventos promocionales o períodos de alta demanda. Esta capacidad de elasticidad horizontal elimina la necesidad de sobredimensionar recursos durante períodos normales, optimizando costos operativos.

La implementación de timeouts y circuit breakers en las integraciones con servicios externos protege la disponibilidad del sistema principal ante fallos de dependencias. Las llamadas a la API de Stripe y el servicio de envío de correos incluyen límites de tiempo de espera que, al excederse, retornan errores controlados en lugar de bloquear indefinidamente la ejecución. Si un servicio externo presenta fallos repetidos, el sistema puede degradar gracefully deshabilitando temporalmente funcionalidades no críticas mientras mantiene operativas las capacidades esenciales de navegación y consulta de productos.

3.4 Controles de Seguridad

El firewall de aplicación web se implementa mediante las políticas de seguridad nativas de Vercel, que filtran automáticamente tráfico malicioso conocido, bloqueando patrones de ataque comunes como escaneos de puertos, intentos de path traversal y solicitudes con payloads excesivamente grandes. La configuración de cabeceras HTTP de seguridad incluye Content-Security-Policy para prevenir inyección de scripts no autorizados, X-Frame-Options para proteger contra clickjacking, y Strict-Transport-Security para forzar el uso de HTTPS en todas las comunicaciones.

El control de acceso basado en roles se materializa a través de la combinación de Row Level Security en PostgreSQL y lógica de autorización en el middleware de Next.js. Las políticas RLS definen qué filas de cada tabla puede leer o modificar un usuario según su rol almacenado en el perfil. Un cliente regular solo puede acceder a sus propias órdenes, mientras que un administrador tiene permisos completos sobre productos y puede visualizar todas las transacciones. Estas políticas se ejecutan a nivel de base de datos, proporcionando una capa de protección adicional incluso si existieran vulnerabilidades en el código de la aplicación.

La protección contra fuerza bruta en intentos de inicio de sesión se implementa mediante el sistema de rate limiting de Supabase Auth, que bloquea temporalmente cuentas después de cinco intentos fallidos consecutivos. El bloqueo tiene una duración de 15 minutos, suficiente para frustrar ataques automatizados pero breve para no impactar severamente a usuarios legítimos que hayan olvidado su contraseña. El sistema envía notificaciones por correo cuando detecta múltiples intentos fallidos, alertando al propietario de la cuenta sobre posibles intentos de acceso no autorizado.

El principio de mínimo privilegio se aplica rigurosamente en la configuración de claves de API y tokens de acceso. El frontend utiliza exclusivamente la clave anónima de Supabase que

solo permite operaciones autorizadas por las políticas RLS, mientras que operaciones administrativas como la eliminación permanente de registros o modificación de roles se ejecutan mediante la Service Role Key que permanece en el servidor. Esta segregación impide que usuarios malintencionados puedan escalar privilegios manipulando solicitudes del cliente.

4. Estrategias de Monitoreo y Seguimiento

4.1 Monitoreo de Rendimiento

El análisis de rendimiento del frontend se implementa mediante Vercel Analytics, una herramienta integrada que captura métricas de Web Vitals sin impactar la experiencia del usuario. El sistema registra el First Contentful Paint que mide cuánto tiempo transcurre hasta que el usuario ve el primer contenido renderizado, el Largest Contentful Paint que indica cuándo el elemento principal de la página se vuelve visible, y el Cumulative Layout Shift que cuantifica cambios inesperados en el diseño durante la carga. Estas métricas permiten identificar páginas con rendimiento subóptimo y priorizar optimizaciones en componentes que impactan la experiencia de usuario.

El tiempo de respuesta de las funciones serverless se monitorea automáticamente por Vercel, registrando la duración de cada invocación y alertando cuando se exceden umbrales configurados. Las Server Actions que realizan operaciones de base de datos intensivas, como la generación de reportes de ventas o la actualización masiva de inventario, se instrumentan con marcadores de tiempo que permiten identificar cuellos de botella específicos. Los logs estructurados capturan no solo la duración total sino también el tiempo consumido en cada operación individual, facilitando la optimización selectiva de las secciones más costosas.

El monitoreo de cachés y revalidación de datos se gestiona mediante las estrategias incorporadas de Next.js que registran hits y misses de cache, permitiendo ajustar las políticas de

revalidación para balancear frescura de datos con rendimiento. Las páginas de productos se configuran con revalidación incremental cada 60 segundos, garantizando que cambios en precios o disponibilidad se reflejen rápidamente sin necesidad de regenerar toda la página en cada visita. El dashboard de Vercel visualiza las tasas de acierto de cache y el volumen de regeneraciones, facilitando el ajuste fino de estas estrategias.

4.2 Monitoreo de Disponibilidad

La supervisión de uptime se implementa mediante health checks automáticos ejecutados por Vercel cada 60 segundos desde múltiples regiones geográficas. Estos checks realizan solicitudes HTTP a endpoints críticos de la aplicación y verifican que retornen códigos de estado exitosos dentro de un tiempo límite de 10 segundos. Cuando se detecta una falla en tres verificaciones consecutivas, el sistema dispara alertas por correo electrónico y webhooks configurados, notificando al equipo de desarrollo para investigación inmediata.

El monitoreo de la base de datos se realiza mediante el panel de control de Supabase que expone métricas de conexiones activas, tasa de queries por segundo y utilización de recursos computacionales. Los gráficos históricos permiten identificar patrones de uso y planificar escalado antes de alcanzar límites de capacidad. Las alertas configurables notifican cuando el pool de conexiones se aproxima a su límite máximo, permitiendo intervención proactiva antes de que nuevos usuarios experimenten errores de conexión.

La trazabilidad de transacciones críticas se implementa mediante logs estructurados en formato JSON que registran cada paso del flujo de compra, desde la adición de productos al carrito hasta la confirmación del pago y envío del correo de confirmación. Cada log incluye un identificador de correlación único que permite seguir una transacción específica a través de todos los sistemas involucrados. En caso de fallo de un pago, estos logs permiten reconstruir

exactamente qué ocurrió, identificando si el problema se originó en la validación del carrito, la comunicación con Stripe o el registro de la orden en la base de datos.

4.3 Monitoreo de Seguridad

El análisis de intentos de acceso no autorizados se realiza mediante logs de autenticación de Supabase que registran todos los intentos de login exitosos y fallidos, incluyendo la dirección IP origen, timestamp y usuario objetivo. Los reportes automáticos generados semanalmente identifican patrones sospechosos como múltiples intentos fallidos desde una misma IP o intentos de acceso a cuentas inexistentes. La correlación de estos eventos con bases de datos de IPs maliciosas conocidas permite bloquear proactivamente fuentes de tráfico malicioso.

La detección de anomalías en patrones de tráfico se habilita mediante análisis de series temporales que establecen líneas base de comportamiento normal para cada endpoint. Los algoritmos de detección identifican desviaciones significativas como incrementos súbitos en solicitudes a endpoints administrativos o picos de tráfico en horarios inusuales. Estas anomalías pueden indicar intentos de scraping, ataques de denegación de servicio o compromisos de cuentas de administrador que requieren investigación inmediata.

El monitoreo de errores de aplicación se implementa mediante la captura centralizada de excepciones no controladas, errores de validación y fallos de integración con servicios externos. Los errores se clasifican por severidad y tipo, permitiendo priorizar la resolución de problemas que afectan funcionalidades críticas como el procesamiento de pagos. Los reportes de tendencias identifican errores recurrentes que pueden indicar bugs sistemáticos o ataques dirigidos a vulnerabilidades específicas.

4.4 Métricas Clave de Negocio

El seguimiento de conversión y abandono del carrito se instrumenta mediante eventos personalizados que registran cuándo un usuario añade productos, procede al checkout y completa la compra. El cálculo de la tasa de abandono identifica en qué paso del flujo los usuarios desisten más frecuentemente, permitiendo optimizar la experiencia en esos puntos críticos. Los embudos de conversión visualizan el porcentaje de visitantes que progresan desde la visualización de productos hasta la confirmación de pago, cuantificando el impacto de cambios en la interfaz sobre las métricas de negocio.

El análisis de productos más visitados y comprados se realiza mediante agregación de eventos de visualización y adición al carrito, identificando qué ítems del catálogo generan mayor interés. La correlación entre visualizaciones y compras revela productos con alta tasa de conversión que podrían promocionarse más agresivamente, así como productos con muchas vistas pero pocas ventas que requieren ajustes de precio o descripción. Los reportes de productos sin ventas durante períodos prolongados facilitan decisiones de rotación de inventario.

El monitoreo de tiempo promedio de sesión y páginas por visita proporciona indicadores de engagement que reflejan el interés de los usuarios en el contenido. Sesiones cortas con pocas páginas visitadas pueden indicar problemas de usabilidad o desalineación entre expectativas y contenido ofrecido. La segmentación de estas métricas por fuente de tráfico identifica canales de adquisición que aportan usuarios de mayor calidad, informando decisiones de asignación de presupuesto de marketing.

5. Herramientas de Monitoreo Implementadas

5.1 Vercel Analytics

Vercel Analytics constituye la herramienta principal de observabilidad del frontend, proporcionando visibilidad completa sobre el rendimiento percibido por los usuarios finales sin

requerir configuración adicional o inclusión de scripts de terceros que podrían impactar la velocidad de carga. La plataforma captura automáticamente las métricas Core Web Vitals establecidas por Google como estándares de calidad web, permitiendo medir objetivamente la experiencia de usuario y su impacto en el posicionamiento en motores de búsqueda.

El dashboard de Analytics expone gráficos históricos de cada métrica segmentados por página, dispositivo y región geográfica, revelando patrones específicos como rendimiento degradado en dispositivos móviles o latencia elevada en regiones distantes de los servidores. Los percentiles 50, 75 y 95 permiten entender no solo el comportamiento promedio sino también la experiencia de los usuarios en el peor caso, fundamentales para identificar optimizaciones que beneficien a toda la base de usuarios.

La integración nativa con el sistema de deployment permite comparar métricas entre versiones de la aplicación, validando que cambios recientes no hayan degradado el rendimiento. Los reportes automáticos alertan cuando nuevos despliegues introducen regresiones significativas en Web Vitals, facilitando rollbacks rápidos antes de que los problemas impacten a la mayoría de usuarios. Esta capacidad de testing sintético continuo reduce el riesgo asociado con iteraciones frecuentes del código.

5.2 Supabase Dashboard

El panel de administración de Supabase proporciona visibilidad operacional sobre la base de datos, sistema de autenticación y almacenamiento de archivos desde una interfaz unificada. El módulo de métricas de base de datos expone el número de conexiones activas, queries ejecutados por segundo, y distribución de tiempos de respuesta, permitiendo identificar consultas lentas que requieren optimización mediante índices adicionales o reestructuración de esquemas.

El explorador de logs de autenticación facilita la auditoría de actividad de usuarios, registrando cada inicio de sesión, cierre de sesión y cambio de contraseña con marcas temporales precisas e información de contexto como dirección IP y user agent. Los filtros avanzados permiten buscar eventos específicos como intentos de acceso a cuentas administrativas o sesiones iniciadas desde ubicaciones geográficas inusuales, acelerando investigaciones de seguridad.

La visualización de uso de almacenamiento muestra el volumen de imágenes y archivos subidos por los administradores, proyectando cuándo se alcanzarán límites del plan actual y permitiendo planificación proactiva de upgrades. Los reportes de tráfico de red hacia el storage bucket identifican archivos frecuentemente descargados que podrían beneficiarse de caching adicional mediante una CDN externa.

5.3 Stripe Dashboard

El dashboard de Stripe complementa el monitoreo de la plataforma proporcionando visibilidad completa sobre el flujo de transacciones financieras, desde intentos de pago iniciales hasta liquidaciones bancarias. Los gráficos de volumen de transacciones y valores procesados permiten correlacionar actividad comercial con eventos de marketing o cambios en la aplicación, cuantificando el retorno sobre inversión de iniciativas específicas.

El explorador de eventos de webhook expone todas las notificaciones enviadas por Stripe a la aplicación, incluyendo timestamp, payload y código de respuesta HTTP retornado. Los webhooks fallidos se destacan visualmente, facilitando la identificación de problemas de integración que podrían resultar en órdenes no confirmadas o inventario no actualizado. La capacidad de reenviar webhooks manualmente permite recuperar transacciones que fallaron por interrupciones temporales de servicio.

Las alertas configurables de Stripe notifican sobre eventos críticos como disputas iniciadas por clientes, fallos repetidos de una tarjeta específica o incrementos súbitos en la tasa de fraude. Estas notificaciones permiten respuesta inmediata a situaciones que podrían impactar negativamente la reputación del comercio o generar pérdidas financieras si no se atienden oportunamente.

5.4 Logs Estructurados

Los logs estructurados en formato JSON emitidos por las funciones serverless se centralizan automáticamente en Vercel Logs, proporcionando un registro cronológico completo de la ejecución de la aplicación. Cada entrada de log incluye metadata estandarizada como nivel de severidad, timestamp ISO 8601, identificador de función y duración de ejecución, facilitando búsquedas y agregaciones mediante queries estructuradas.

La implementación de niveles jerárquicos de logging permite ajustar dinámicamente la verbosidad de los logs según el entorno de ejecución. En producción se registran únicamente errores y advertencias para minimizar costos de almacenamiento, mientras que en ambientes de desarrollo se capturan logs de nivel debug que incluyen detalles de cada query de base de datos ejecutado y respuestas completas de APIs externas. Esta flexibilidad acelera el debugging sin comprometer el rendimiento en producción.

Los logs de errores incluyen stack traces completos y contexto relevante como parámetros de entrada que causaron la excepción, facilitando la reproducción y resolución de bugs reportados por usuarios. La retención configurable de logs permite mantener históricos prolongados para análisis forense de incidentes, mientras que logs rutinarios se descartan después de períodos breves para controlar costos de almacenamiento.

6. Referencias

- Institute of Electrical and Electronics Engineers. IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society Press, 1998. IEEE Std 830-1998.
- National Institute of Standards and Technology. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. NIST, 2018. Disponible en https://www.nist.gov/cyberframework
- Open Web Application Security Project. OWASP Top Ten Web Application Security Risks. OWASP Foundation, 2021. Disponible en https://owasp.org/www-project-top-ten/
- Vercel Inc. Next.js Documentation: Security Best Practices. Vercel, 2024. Disponible en https://nextjs.org/docs/app/building-your-application/configuring/security
- Supabase Inc. PostgreSQL Row Level Security Documentation. Supabase, 2024. Disponible en https://supabase.com/docs/guides/auth/row-level-security
- Stripe Inc. Stripe Security and Compliance Documentation. Stripe, 2024. Disponible en https://stripe.com/docs/security
- Center for Internet Security. CIS Controls Version 8. Center for Internet Security, 2021. Disponible en https://www.cisecurity.org/controls