

Preguntas frecuentes

Respuestas técnicas precisas. Sin marketing-speak.

Producto

¿Qué es exactamente el AST parsing y por qué es mejor que un allowlist?

Un allowlist de operaciones bloquea por el primer token de la query: si empieza con DELETE, bloquea. Si empieza con SELECT, permite. Es un instrumento sin granularidad.

El AST (Abstract Syntax Tree) parsing construye el árbol sintáctico completo de la query. Vericto recorre ese árbol buscando nodos que violen las reglas. Esto permite detectar:

- `UPDATE products SET price = 0` sin WHERE clause (UpdateStmt > WhereClause = NULL)
- `DELETE FROM users` sin WHERE en cualquier nivel de anidamiento
- `TRUNCATE` disfrazado dentro de un CTE
- `DELETE FROM orders LIMIT 0`, un DELETE que no borra nada pero es sintácticamente peligroso

Un allowlist no puede detectar ninguno de estos casos porque todos empiezan con tokens que parecen seguros en superficie.

¿De qué formas puedo conectar Vericto?

De cinco formas, en dos momentos:

- **Proxy TCP** (runtime): apuntas tu connection string al proxy en lugar de a tu base de datos. Intercepta cada query en el protocolo (Postgres/MySQL) y bloquea en línea. Cero cambios de código.
- **Runtime API** (runtime): tu aplicación envía cada query a Vericto antes de ejecutarla, una por llamada, y la ejecuta solo si la respuesta es ALLOWED. Útil para Oracle y SQL Server, o cuando no puedes cambiar el connection string.
- **CLI** (antes de ejecutar): el binario `vericto` valida SQL en pre-commit y CI/CD, con un exit code según el veredicto.
- **Validation API** (antes de ejecutar): envías archivos o lotes de SQL por HTTP y recibes el veredicto de cada sentencia, sin instalar nada.
- **MCP** (antes de ejecutar): agentes de IA como Claude Code o Kiro validan el SQL mientras lo escriben.

Todas están disponibles en todos los planes, incluido el gratuito. Consulta [Integraciones](#) para elegir.

¿Tengo que modificar mi código de aplicación para usar Vericto?

No. Vericto actúa como proxy TCP transparente. Solo cambias el host en tu connection string:

```
- DATABASE_URL=postgres://user:pass@prod-db.host:5432/db
+ DATABASE_URL=postgres://user:pass@localhost:5433/db
```

Tu ORM (SQLAlchemy, Prisma, ActiveRecord, Drizzle) no sabe que hay un proxy en el medio. Las queries llegan exactamente igual. Vericto las intercepta, las analiza, y las reenvía si son seguras.

¿Vericto soporta bases de datos NoSQL como MongoDB o Redis?

No. Vericto es específico para bases de datos SQL. El AST parsing es una técnica que aplica al lenguaje SQL: extenderlo a NoSQL requeriría parsers completamente diferentes y el concepto de "query destructiva" es menos definido en esos contextos.

El foco actual es SQL con los dialectos más usados en producción con agentes de IA: Postgres, MySQL, Oracle y SQL Server, con SQLite y Snowflake en el roadmap. Esto cubre el 95%+ de los casos de uso de LLM-to-SQL en producción.

Rendimiento y disponibilidad

¿Cuánto overhead añade Vericto a mis queries?

Depende del modo de integración:

- **Proxy TCP (recomendado):** la evaluación ocurre dentro del proxy, en tu red, sin llamadas externas. En nuestras pruebas añade menos de 1 ms por query, también en queries medianas y complejas.
- **Runtime API (HTTP):** cada query viaja a la API de Vericto para evaluarse antes de ejecutarla, así que el tiempo incluye la red. En nuestras pruebas el promedio es de unos 16 ms por evaluación.

El proxy está escrito en Rust con tokio para manejar conexiones TCP de forma asíncrona, sin pausas de garbage collection. El registro de eventos en el audit trail es asíncrono y queda fuera del camino crítico de la query.

¿Qué pasa si Vericto tiene un downtime? ¿Mi aplicación deja de funcionar?

El proxy TCP corre en tu infraestructura y lo operas tú, con tus propios controles de seguridad y tu gestión de secretos. Vericto aloja el control plane: dashboard, reglas y audit trail.

Si el control plane no está disponible, tu aplicación sigue funcionando: el proxy continúa evaluando cada query con el último ruleset y la última política sincronizados, y toma los cambios cuando el servicio vuelve.

Recomendamos ejecutar el proxy con `VERICTO_TELEMETRY_BUFFER=disk`: así los eventos del audit trail se guardan en disco durante la caída, sobreviven a reinicios y se envían al recuperarse.

Puedes ver el estado del servicio en vericto.com/status.

Seguridad y datos

¿Vericto almacena el contenido de mis queries SQL?

Depende del modo de telemetría que elijas en el dashboard:

- **Raw** (por defecto): se guarda el texto de cada query en el audit trail, cifrado en reposo con AES-256-GCM. Te da el máximo detalle forense y la explicación de cada bloqueo.
- **Sanitized**: Vericto nunca guarda los valores reales. Los literales se reemplazan por marcadores antes de almacenar la query, y en el dashboard ves la sentencia ofuscada, por ejemplo `DELETE FROM users WHERE id = $1`, y no la query real que se ejecutó. Con el proxy TCP y la CLI el reemplazo ocurre en tu red, así que los valores ni siquiera salen de ella. Con la Runtime API, la Validation API y MCP, la query viaja a Vericto para evaluarse y se sanitiza antes de guardarse.

El acceso al audit trail requiere autenticación con permisos del workspace. La retención depende del plan: 7 días (Free), 30 días (Builder), 90 días (Team) y configurable en Enterprise. Puedes exportar y purgar el audit trail cuando quieras.

Los datos de las queries nunca se usan para entrenar modelos ni se comparten con terceros.

¿Vericto sirve para requisitos enterprise y compliance?

Sí. Vericto se despliega en tu propia infraestructura (tus credenciales nunca salen de tu red) e incluye las piezas que exige un entorno enterprise:

- **Audit trail firmado con Ed25519**, verificable offline: evidencia directa para SOC2 (CC6.1, CC6.3, CC7.2) e ISO 27001.
- **SSO / OIDC** y control de acceso por roles para tu equipo.
- **Reglas custom en YAML** para políticas específicas de tu organización.

El plan Enterprise añade despliegue asistido, SLA y soporte dedicado.

¿Vericto tiene certificación SOC 2 o ISO 27001?

Vericto está en proceso de certificación SOC 2. Todavía no contamos con un informe SOC 2 ni con la certificación ISO 27001, y lo publicaremos aquí cuando esté disponible.

Lo que sí te da Vericto hoy es **evidencia para tus propias auditorías**: un audit trail de cada decisión, exportable en CSV/JSON con firma Ed25519 verificable sin conexión, y reportes de validación firmados. Para revisiones de seguridad, escríbenos a security@vericto.com.

¿Vericto firma un acuerdo de tratamiento de datos (DPA)?

Sí. El [Acuerdo de Tratamiento de Datos](#) forma parte de los Términos de Servicio y aplica a todos los clientes desde que los aceptan: Vericto actúa como encargado y tu organización como responsable. Cubre GDPR, Ley 1581, LGPD y la LFPDPPP, e incluye los subencargados, las medidas de seguridad y las transferencias internacionales.

Si tu área de compras necesita una copia firmada, descarga el PDF desde vericto.com/dpa, complétalo y envíalo a legal@vericto.com.

¿Cómo funciona el modo dry-run para CI/CD?

El CLI de Vericto permite validar un archivo de queries SQL contra el ruleset de tu workspace sin necesidad de conexión a la base de datos:

```
vericto check migrations.sql --dialect postgres
```

✓ 12 queries analizadas.

✗ 1 query destructiva detectada:

Line 47: TRUNCATE TABLE orders

Rule: VERICTO-011 (severity: CRITICAL)

AST node: TruncateStmt > relation: orders

Suggested fix: DELETE FROM orders WHERE created_at < NOW() - INTERVAL
'90 days'

El proceso termina con exit code 1 si hay queries destructivas (el pipeline de CI falla) o exit code 0 si todo está limpio. Compatible con GitHub Actions, GitLab CI y Jenkins.

¿No encuentras tu respuesta? Escríbenos a hello@vericto.com y te respondemos en menos de 1 día hábil.