

WHITE PAPER

Modernización de Aplicaciones Legacy Impulsada por IA

Cómo la IA Generativa Transforma la Migración de Código desde Sistemas Legacy a Arquitecturas Modernas

Preparado por: **BotMan AI**

Producto: **Codacle — Application Modernization Platform**

Fecha: **Marzo 2026**

Versión: **1.1**

1. Resumen Ejecutivo

Las organizaciones de todo el mundo están atrapadas en una paradoja costosa. Las aplicaciones legacy que sustentan sus procesos de negocio más críticos —construidas hace décadas en tecnologías como COBOL, Delphi, Oracle Forms y Visual Basic— se vuelven cada vez más costosas de mantener, difíciles de asegurar e imposibles de evolucionar. La investigación de la industria muestra consistentemente que las empresas destinan entre el 60 y el 80 por ciento de sus presupuestos de TI simplemente a mantener estos sistemas envejecidos en funcionamiento, dejando una fracción para la innovación y el crecimiento.

El mercado de modernización de aplicaciones legacy refleja esta urgencia. Valorado en aproximadamente USD 25 mil millones en 2025, se proyecta que crecerá a una tasa de crecimiento anual compuesta superior al 15 por ciento hasta finales de la década. Las organizaciones en servicios financieros, atención médica, retail, manufactura y gobierno están compitiendo para reemplazar arquitecturas monolíticas y obsoletas con sistemas modernos, nativos de la nube, que puedan admitir microservicios, diseño API-first y procesamiento de datos en tiempo real.

Sin embargo, los enfoques tradicionales de modernización —reescrituras manuales llevadas a cabo por grandes equipos de consultoría en plazos de varios años— siguen siendo lentos, costosos y propensos al fracaso. La lógica de negocio crítica se pierde en la traducción, los plazos se extienden más allá de las estimaciones iniciales y el código resultante frecuentemente no alcanza los estándares de calidad del sistema original.

Codacle, desarrollado por BotMan AI, ofrece un enfoque fundamentalmente diferente. Al aprovechar la IA Generativa a lo largo de todo el ciclo de modernización —desde el análisis del código legacy y la construcción del grafo de conocimiento hasta el diseño de arquitectura y la generación automatizada de código— Codacle reduce el tiempo y costo de modernización

entre un 50 y un 90 por ciento, produciendo código limpio, mantenible y bien documentado en stacks tecnológicos modernos.

Este white paper examina el desafío de las aplicaciones legacy, presenta la metodología de Codacle, expone resultados reales de migraciones a escala empresarial y explica cómo la modernización impulsada por IA está redefiniendo lo que las organizaciones pueden esperar de sus iniciativas de transformación.

2. El Desafío de las Aplicaciones Legacy

2.1 La Magnitud del Problema

Los sistemas legacy son mucho más omnipresentes de lo que muchos ejecutivos perciben. Las aplicaciones construidas en COBOL, Delphi, Visual Basic, Oracle Forms y tecnologías similares continúan procesando transacciones, gestionando cadenas de suministro y atendiendo clientes en prácticamente todos los sectores. Estos sistemas frecuentemente estaban bien diseñados para su época, pero el panorama tecnológico ha cambiado drásticamente desde que fueron desplegados.

La carga financiera es asombrosa. La investigación de múltiples fuentes de la industria indica que las grandes empresas destinan rutinariamente entre el 60 y el 80 por ciento de sus presupuestos totales de TI al mantenimiento y operación de sistemas legacy. El gobierno federal de los EE.UU., por ejemplo, ha destinado históricamente alrededor del 80 por ciento de su gasto en TI a operaciones y mantenimiento de infraestructura envejecida, dejando solo el 20 por ciento para desarrollo y modernización. En el sector privado, se estima que la deuda técnica por sí sola consume el 40 por ciento de los presupuestos de TI, y esta cifra se compone aproximadamente a un 20 por ciento anual cuando no se aborda.

La crisis de talento agrava la presión financiera. El grupo de desarrolladores con habilidades en lenguajes legacy se reduce rápidamente a medida que los profesionales experimentados se jubilan. Las universidades han eliminado en gran medida COBOL y lenguajes similares de sus planes de estudio. Las organizaciones se encuentran compitiendo por un grupo de talentos cada vez más pequeño, pagando salarios premium por una experiencia que se vuelve más escasa cada año.

2.2 Riesgos y Costos de la Inacción

Las consecuencias de retrasar la modernización van mucho más allá de los excesos presupuestarios. Las organizaciones que se aferran a los sistemas legacy enfrentan tres categorías interconectadas de riesgo:

- **Riesgo de mantenimiento y operacional.** Las aplicaciones legacy frecuentemente carecen de documentación adecuada. Los desarrolladores que las construyeron originalmente a menudo han abandonado la organización, llevando consigo el conocimiento institucional crítico. Cada modificación conlleva el riesgo de efectos secundarios no deseados en bases de código monolíticas estrechamente acopladas. Las tareas de mantenimiento rutinario que tomarían horas en un sistema moderno pueden consumir días o semanas.
- **Riesgo de inflexibilidad y competitividad.** Las arquitecturas monolíticas no pueden adaptarse rápidamente a los nuevos requisitos de negocio. La integración con APIs modernas, el soporte de interfaces móviles o el despliegue en entornos de nube

requieren soluciones alternativas extensas. Los competidores que operan en stacks modernos pueden lanzar nuevas funcionalidades en días, mientras que las organizaciones dependientes de sistemas legacy luchan con ciclos de lanzamiento medidos en meses.

- **Riesgo de seguridad y cumplimiento normativo.** Los sistemas legacy son objetivos principales para los ciberataques porque frecuentemente se ejecutan en software sin soporte que ya no recibe parches de seguridad. Los mecanismos de autenticación obsoletos, los flujos de datos no cifrados y las arquitecturas rígidas dificultan el cumplimiento de regulaciones en evolución como GDPR, HIPAA, PCI DSS y SOX. Las instituciones financieras, por ejemplo, destinan una proporción desproporcionada de sus presupuestos de TI únicamente al mantenimiento del cumplimiento normativo.

2.3 Por Qué la Migración Tradicional es Insuficiente

Las organizaciones que se comprometen con la modernización frecuentemente descubren que el enfoque tradicional —reunir grandes equipos de desarrolladores para reescribir aplicaciones manualmente— introduce su propio conjunto de problemas.

Las reescrituras manuales son lentas. Un proyecto de migración empresarial típico abarca de dos a cinco años, durante los cuales el sistema legacy debe seguir operando y el negocio debe gestionar entornos paralelos. Son costosas, excediendo frecuentemente las estimaciones presupuestarias iniciales a medida que emerge la complejidad oculta. Lo más crítico es que son riesgosas: los desarrolladores que trabajan con documentación incompleta y conocimiento fragmentado del sistema original rutinariamente omiten casos límite, interpretan incorrectamente las reglas de negocio o introducen inconsistencias que solo salen a la luz en producción.

Las herramientas de asistencia de código —la generación actual de asistentes de desarrollo potenciados por IA— ayudan a los desarrolladores individuales a escribir código más rápido, pero no resuelven el desafío fundamental. Carecen de la capacidad de comprender el contexto completo de una aplicación legacy: las relaciones entre cientos de procedimientos, funciones, tablas y reglas de negocio que juntos definen cómo opera el sistema. Sin esta comprensión holística, la modernización sigue siendo un proceso fragmentado y propenso a errores.

Lo que las organizaciones necesitan es un enfoque que combine la comprensión contextual profunda de toda la base de código legacy con la velocidad y consistencia de la automatización. Esta es precisamente la brecha que Codacle fue diseñado para llenar.

3. El Enfoque de Codacle: Modernización Impulsada por IA

Codacle es la plataforma propietaria de BotMan AI para acelerar la modernización de aplicaciones legacy a través de la IA Generativa. A diferencia de las herramientas convencionales que asisten a los desarrolladores en tareas de codificación aisladas, Codacle opera a nivel de aplicación —analizando, comprendiendo y transformando sistemas completos desde su stack tecnológico original a arquitecturas modernas.

El principio fundamental detrás de Codacle es sencillo: mapear en detalle cada paso que un equipo humano especializado realizaría si condujera la migración manualmente, y luego desplegar aceleradores basados en IA que reproducen cada paso de forma autónoma, preservando la capacidad de los expertos humanos para validar y ajustar en cada etapa. Esto se implementa a través de un pipeline propietario de más de 60 pasos deterministas, cada uno

potenciado por IA Generativa con exactamente el contexto adecuado del grafo de conocimiento —ni más, ni menos.

3.1 Construcción del Grafo de Conocimiento

La base del enfoque de Codacle es un grafo de conocimiento integral que captura la estructura completa, el comportamiento y las relaciones dentro de la aplicación legacy.

Durante la fase de análisis, Codacle descompone el código fuente y la base de datos en sus elementos constituyentes —paquetes, clases, métodos, funciones, procedimientos almacenados, tablas, vistas y triggers— de acuerdo con las convenciones del lenguaje de programación original. Cada elemento recibe documentación detallada tanto en lenguaje natural como en pseudocódigo, generada automáticamente por la IA a partir del propio código fuente.

De manera crítica, el grafo de conocimiento también mapea todas las relaciones entre elementos: qué métodos pertenecen a qué clases, qué métodos llaman a otros métodos, qué procedimientos leen o escriben en qué tablas, cómo fluyen los datos entre componentes y cómo las interacciones de los usuarios desencadenan cadenas de operaciones. Este mapeo relacional es lo que distingue a Codacle de las herramientas de análisis de código más simples. Proporciona a la IA una representación estructurada y semánticamente rica de toda la aplicación —no solo archivos o funciones individuales, sino la red completa de dependencias e interacciones que definen el comportamiento del sistema.

Los grafos de conocimiento proporcionan una forma estructurada de comprender las relaciones entre entidades, lo que conduce a una mejor comprensión contextual y reduce significativamente el riesgo de alucinaciones de la IA. Al tener mapeadas todas las conexiones entre elementos, ninguna información relevante queda excluida durante el proceso de transformación.

3.2 Diseño de Arquitectura Objetivo

Con el grafo de conocimiento establecido, Codacle pasa al diseño de arquitectura. En lugar de aplicar una plantilla genérica, la plataforma analiza el grafo para identificar los módulos de negocio más relevantes y genera una nueva arquitectura de microservicios centrada en esos módulos.

La arquitectura objetivo presenta una separación limpia entre frontend y backend, con APIs RESTful que definen el contrato entre capas. Codacle admite una amplia variedad de stacks tecnológicos modernos para adaptarse a las preferencias e infraestructura existente de cada organización:

- **Tecnologías frontend:** React, Angular y otros frameworks modernos de JavaScript.
- **Tecnologías backend:** Java (Spring Boot), .NET 8, Node.js y otras plataformas de nivel empresarial. Otros stacks disponibles bajo solicitud.

Esta fase de diseño de arquitectura incluye un checkpoint de revisión deliberado. Antes de que se genere cualquier código, expertos humanos —de BotMan AI y opcionalmente de la

organización cliente— revisan la arquitectura propuesta para asegurarse de que se alinea con los principios de diseño del cliente, las restricciones de infraestructura y la estrategia tecnológica a largo plazo.

3.3 Generación Automatizada de Código

La generación de código es donde el grafo de conocimiento de Codacle entrega su mayor ventaja. Dado que el grafo conecta cada tabla de base de datos con todo el código de la aplicación original que la lee o escribe, la IA puede comprender y replicar la lógica de negocio original dentro de la nueva estructura arquitectónica.

Generación de backend. Codacle genera automáticamente la capa de backend completa en el lenguaje y framework objetivo. En una migración a Spring Boot, por ejemplo, esto incluye controladores que exponen los endpoints de la API RESTful, servicios que implementan la lógica de negocio, entidades que modelan el dominio de datos y repositorios que gestionan el acceso a datos. Cada componente generado encapsula las reglas de negocio que anteriormente estaban embebidas en los procedimientos y funciones legacy, pero ahora organizadas según las prácticas modernas de ingeniería de software.

Generación de frontend. Codacle utiliza agentes de IA Generativa especializados para analizar el diseño, la disposición y el comportamiento de las pantallas de la aplicación original, y luego genera interfaces de usuario equivalentes en el framework frontend objetivo. La IA separa el código original de cada interacción de usuario en comportamientos de front-end (que se replican a nivel de interfaz) y comportamientos de back-end (que se reemplazan con llamadas a las APIs recién creadas). El resultado es una interfaz moderna y responsiva que preserva el flujo funcional que los usuarios esperan, aprovechando al mismo tiempo las capacidades de la interfaz de usuario contemporánea.

3.4 Validación con Supervisión Humana

Codacle no es una herramienta de automatización de caja negra. En cada etapa del proceso, los expertos humanos pueden revisar, ajustar y validar el resultado generado por la IA:

- **Revisión de arquitectura:** Antes de que comience la generación de código, la arquitectura de microservicios propuesta y el diseño de APIs se revisan contra los estándares y mejores prácticas del cliente.
- **Revisión de calidad del código:** El código generado se inspecciona para verificar su corrección, legibilidad y adherencia a los estándares de codificación. Los ajustes pueden ser realizados tanto por los ingenieros de BotMan AI como por los desarrolladores del cliente.
- **Pruebas asistidas:** Codacle aprovecha las herramientas de IA Generativa para facilitar y acelerar el desarrollo de casos de prueba y automatizar su ejecución contra la solución migrada, asegurando la paridad funcional con el sistema original.
- **Validación automatizada:** Un pipeline de validación autónomo revisa cientos de puntos de datos del código fuente original para validar la precisión del código migrado, dejando evidencia de cada validación.

Este modelo de supervisión humana garantiza que las ventajas de velocidad de la automatización de IA no se produzcan a expensas de la calidad o la alineación con los requisitos de negocio.

3.5 Barbara: Asistente de Código Inteligente

Como parte de la plataforma Codacle, BotMan AI ofrece Barbara —un agente potenciado por IA que recorre el grafo de conocimiento en tiempo real para responder consultas complejas sobre la base de código. Los equipos de desarrollo pueden consultar a Barbara sobre los flujos de información entre componentes, las relaciones entre objetos, los elementos no utilizados o redundantes, y qué componentes deberían modificarse para implementar una nueva funcionalidad.

Barbara se integra directamente en el IDE del desarrollador como una herramienta MCP, siendo compatible con Visual Studio Code, Cursor, Windsurf y otros entornos de desarrollo modernos. Actúa como una capa de documentación viva que hace que el grafo de conocimiento sea accesible y accionable para los equipos de desarrollo —tanto durante el proyecto de migración como mucho después de su finalización.

4. Casos de Uso y Capacidades

El enfoque de Codacle impulsado por IA se aplica a una amplia gama de escenarios de modernización. Si bien la migración completa de aplicaciones es el engagement más común, las capacidades del grafo de conocimiento y la generación de código de la plataforma se extienden a varios casos de uso adyacentes.

Caso de Uso	Descripción
Migración Completa de Aplicaciones	Migración end-to-end de aplicaciones legacy desde tecnologías como Delphi, Visual Basic, COBOL y Oracle Forms a arquitecturas modernas, incluyendo Spring Boot, .NET 8, Node.js, React y Angular.
Documentación Técnica y Funcional	Generación automatizada de documentación integral a partir del código fuente, abarcando la arquitectura de la aplicación, la lógica de negocio, los flujos de datos y los puntos de integración.
Catálogo de Datos Inteligente (Metron)	A través del módulo Metron, creación automatizada de diccionarios de datos enriquecidos a partir de esquemas de base de datos, datos de muestra, logs de consultas y opcionalmente el código fuente de la aplicación, con detección de PII y preparación para el cumplimiento normativo.

4.1 Metron: Acelerando la Gobernanza de Datos

Metron es un módulo especializado dentro del ecosistema de Codacle diseñado para acelerar las iniciativas de gobernanza de datos. Genera automáticamente diccionarios de datos enriquecidos combinando múltiples fuentes de conocimiento: esquemas de base de datos (DDL), datos de muestra, logs de consultas SQL, procedimientos almacenados y opcionalmente el propio grafo de conocimiento de la aplicación.

Para cada campo de la base de datos, Metron produce una descripción, clasifica la sensibilidad de la información (incluyendo la detección de PII directa e indirecta) y proporciona evidencia y

un grado de certeza para cada descripción generada. Cuando el grafo de conocimiento de la aplicación está disponible, la precisión puede alcanzar entre el 90 y el 95 por ciento, porque la IA comprende no solo la estructura de la base de datos, sino también cómo la aplicación lee, escribe y transforma los datos.

Las salidas de Metron están diseñadas para integrarse con plataformas de gobernanza como Collibra, Alation, Informatica EDC, Atlan y dbt Docs, con opciones de exportación que incluyen YAML, CSV y API. El módulo transforma lo que tradicionalmente requiere semanas de esfuerzo manual por parte de los data stewards en un proceso que se completa en horas, produciendo documentación viva que puede actualizarse automáticamente a medida que el esquema de la base de datos evoluciona.

5. Caso de Estudio: Migración Delphi a Escala Empresarial

5.1 Perfil del Cliente

Forus es un retailer multinacional de moda que opera en Chile, Perú, Colombia y Uruguay, con más de 400 tiendas físicas. Como muchos retailers establecidos, Forus había construido un patrimonio tecnológico sustancial a lo largo de décadas, con aplicaciones de negocio principales desarrolladas en Delphi.

5.2 El Desafío

Forus necesitaba modernizar 500 formularios Delphi y 70 procesos batch —que comprenden 550.000 líneas de código acumuladas durante 32 años— migrándolos a una arquitectura moderna basada en Spring Boot para el backend y React para el frontend. La escala del proyecto —que abarcaba la gama completa de operaciones de retail, desde la gestión de inventario hasta la integración con e-commerce— hacía que una reescritura manual tradicional fuera impracticable dentro de plazos y presupuestos aceptables. Se estimaba que una reescritura manual tomaría 16 años.

5.3 El Enfoque de Codacle

BotMan AI desplegó Codacle para ejecutar la migración a través de su metodología de cuatro fases:

1. **Construcción del grafo de conocimiento.** Codacle analizó la base de código Delphi completa y las bases de datos de soporte, construyendo un grafo integral de todos los elementos de la aplicación —procedimientos, funciones, formularios, tablas, procedimientos almacenados— y sus relaciones. Cada elemento fue documentado en lenguaje natural y pseudocódigo.
2. **Diseño de arquitectura.** Usando el grafo de conocimiento, Codacle identificó los módulos de negocio principales y diseñó una nueva arquitectura de microservicios con APIs RESTful. La arquitectura fue revisada y aprobada por el equipo técnico de Forus antes de proceder a la generación de código.
3. **Generación de código.** Codacle generó el backend de Spring Boot —controladores, servicios, entidades y repositorios— replicando la lógica de negocio de los procedimientos Delphi originales dentro de la nueva estructura de API. Los agentes de frontend luego analizaron los formularios Delphi originales y generaron interfaces React equivalentes, conectadas a los nuevos servicios de backend.

4. **Validación y pruebas.** Los expertos humanos revisaron el código generado en cada etapa. Se utilizaron herramientas de IA Generativa para desarrollar casos de prueba y automatizar su ejecución, asegurando la paridad funcional entre las aplicaciones original y modernizada.

5.4 Resultados

La migración impulsada por Codacle entregó resultados medibles en cada dimensión relevante para las partes interesadas empresariales:

- **Reducción de tiempo y costo:** La migración se completó en 14 meses con un 50 a 90 por ciento menos de tiempo y costo en comparación con lo que habría requerido una reescritura manual tradicional —un proyecto que se había estimado en 16 años.
- **Calidad del código:** El código generado fue limpio, legible y mantenible, siguiendo las mejores prácticas modernas de desarrollo y acompañado de documentación integral.
- **Arquitectura moderna:** Forus obtuvo un sistema basado en microservicios que permite el escalado independiente de componentes, la integración directa con nuevas tecnologías y el desarrollo ágil de funcionalidades.
- **Riesgo reducido:** La combinación de automatización impulsada por IA, checkpoints de validación estructurados y plazos comprimidos redujo significativamente los riesgos típicamente asociados con proyectos de migración a gran escala.
- **Documentación automática:** Codacle generó documentación detallada para la nueva base de código, reduciendo la carga de transferencia de conocimiento y haciendo que el sistema sea inmediatamente accesible para los equipos de desarrollo.

50–90%

Reducción en tiempo y costo de migración en comparación con la reescritura manual tradicional

6. Seguridad y Protección de Datos

Las organizaciones que consideran la modernización impulsada por IA tienen razón en querer comprender cómo se gestionará su código fuente y sus datos durante todo el proceso. BotMan AI ha diseñado sus prácticas de seguridad para satisfacer las expectativas de los clientes empresariales en industrias reguladas.

6.1 Almacenamiento y Manejo del Código

Todos los procesos se ejecutan en redes y computadoras seguras. El código fuente del cliente se almacena exclusivamente en estos sistemas seguros y en carpetas de Microsoft OneDrive durante el período del proyecto. Al finalizar el proyecto, todas las copias se eliminan.

permanentemente del almacenamiento de BotMan AI. En ningún momento el código fuente se comparte con partes no autorizadas ni se retiene más allá del plazo acordado del proyecto.

6.2 Uso de IA y LLMs

El uso de modelos de lenguaje de gran escala de Codacle se restringe a APIs de nivel empresarial proporcionadas por Azure, AWS, Google Cloud y Anthropic. Estos proveedores ofrecen compromisos de seguridad empresarial que incluyen cifrado de datos en tránsito y en reposo, sin uso de datos del cliente para el entrenamiento de modelos, y certificaciones de cumplimiento normativo relevantes para industrias reguladas. Si un cliente requiere la exclusión de algún proveedor específico, BotMan AI puede acomodar esta restricción.

BotMan AI no utiliza actualmente LLMs locales o de código abierto para trabajos de modernización. Las pruebas internas han demostrado que los modelos basados en API de nivel empresarial ofrecen una calidad sustancialmente superior para las tareas complejas de razonamiento y generación de código que la modernización exige.

6.3 Autenticación y Autorización

Para las aplicaciones migradas, el enfoque predeterminado de BotMan AI es mantener el mismo método de autenticación y autorización utilizado por la aplicación legacy, asegurando la continuidad para los usuarios finales y minimizando las interrupciones. Alternativamente, el equipo puede integrarse con estándares modernos de autenticación como Active Directory, o construir mecanismos de autenticación completamente nuevos adaptados a los requisitos de seguridad del cliente.

6.4 Acceso a Datos

Las migraciones de código no requieren acceso a datos de bases de datos de producción. BotMan AI solo necesita acceso a datos de prueba o sintéticos durante la fase de pruebas de la migración. Esta separación garantiza que los datos de producción sensibles permanezcan dentro del perímetro de seguridad del cliente durante todo el engagement.

7. Conclusión y Próximos Pasos

La modernización de aplicaciones legacy ya no es opcional para las organizaciones que quieren mantenerse competitivas, seguras y en cumplimiento normativo. Los costos de mantener sistemas envejecidos —medidos en consumo presupuestario, exposición a riesgos de seguridad, escasez de talento y pérdida de agilidad— se acumulan con cada año de retraso.

Codacle transforma lo que tradicionalmente ha sido un esfuerzo de múltiples años, alto riesgo y alto costo en un proceso estructurado y acelerado por IA que entrega resultados en una fracción del tiempo. Al construir un grafo de conocimiento completo de la aplicación legacy, diseñar una arquitectura objetivo moderna y generar código limpio tanto en capas de frontend como de backend —todo con supervisión de expertos humanos en cada etapa— Codacle aborda las causas raíz del fracaso en la modernización: comprensión incompleta, ejecución lenta y calidad inconsistente.

Los resultados hablan por sí mismos: reducciones del 50 al 90 por ciento en tiempo y costo, calidad de código de nivel empresarial, documentación integral y arquitecturas modernas que posicionan a las organizaciones para años de innovación futura.

Para Comenzar

BotMan AI ofrece un camino estructurado desde la evaluación inicial hasta la modernización a escala completa:

- **Sesión de descubrimiento.** Una sesión colaborativa para evaluar su portfolio de sistemas legacy, identificar las aplicaciones prioritarias, definir las arquitecturas objetivo y estimar el alcance y los plazos.
- **Prueba de concepto.** Una migración piloto enfocada de una aplicación o módulo representativo, demostrando la metodología de Codacle y produciendo resultados medibles en semanas.
- **Engagement a escala completa.** Un programa de modernización estructurado que abarca su portfolio completo de aplicaciones, con hitos claros, checkpoints de calidad y colaboración continua entre BotMan AI y su equipo técnico.

¿Listo para modernizar sus aplicaciones legacy?

Contáctenos para programar una sesión de descubrimiento y ver Codacle en acción.

Este documento es confidencial y propiedad de BotMan AI. Está destinado al uso exclusivo del destinatario y no puede ser reproducido, distribuido ni divulgado a terceros sin el consentimiento previo por escrito de BotMan AI.