



BIZNET DOCMAIN

CODE INTELLIGENCE

Propuesta técnico-comercial

Entendimiento, evolución y modernización de aplicaciones críticas

Mensaje central

DOCMAIN Code Intelligence convierte el código fuente y los artefactos de software en una fuente de verdad del negocio digital.

No parte prometiendo migrar automáticamente: primero entiende, documenta, visualiza, analiza impacto y gobierna el cambio con evidencia.

Preparado por BIZNET
Gestión Documental Cognitiva y Agentic

Contenido

- 1. Resumen ejecutivo**
- 2. El problema que resuelve**
- 3. Propuesta de valor**
- 4. El código como fuente de verdad**
- 5. Arquitectura técnica por capas**
- 6. Capa agentic: agentes especializados**
- 7. Casos de uso principales**
 - 7.1 Migración de sistemas legados
 - 7.2 Cambios evolutivos
 - 7.3 Visualización del negocio digital
 - 7.4 Auditoría, cumplimiento y continuidad
- 8. Flujo técnico de trabajo**
- 9. Relación con DevOps, Git y CI/CD**
- 10. Extensión hacia infraestructura y operaciones**
- 11. Entregables de una primera implementación**
- 12. Posicionamiento comercial**
- 13. Conclusión**

1. Resumen ejecutivo

BIZNET DOCMAIN Code Intelligence es una plataforma **agentic** que transforma código fuente, aplicaciones, bases de datos, APIs, jobs, scripts, documentación técnica, tickets y evidencias operacionales en una **fuentes de verdad del negocio digital**. Su valor no se limita a migrar sistemas: permite que la organización entienda, con evidencia, cómo funcionan realmente sus aplicaciones críticas.

La pregunta que responde no es solo «¿cómo muevo este código?», sino las cuatro que la anteceden y que casi nunca tienen respuesta documentada: **qué hace cada sistema, qué reglas de negocio contiene, qué depende de qué y qué impacto real tendría modificarlo**. Sobre esa base, DOCMAIN habilita decidir la estrategia correcta para evolucionar, modernizar o migrar de forma segura.

Principio de diseño

Antes de cambiar, evolucionar o migrar una aplicación crítica, primero hay que entenderla.

La promesa es deliberadamente honesta. No prometemos migrar millones de líneas automáticamente. Prometemos entender, documentar y gobernar millones de líneas de código para decidir, con evidencia, qué evolucionar, modernizar, encapsular, migrar, mantener o retirar.

2. El problema que resuelve

En muchas organizaciones, el conocimiento real del negocio no está en un manual: está embebido en el código. La documentación puede estar incompleta, los expertos que lo escribieron pueden haberse ido, y las dependencias entre componentes rara vez están completamente mapeadas. El resultado es una dependencia silenciosa pero crítica de sistemas que pocos comprenden por completo.

El problema real: el negocio vive en el código

El conocimiento crítico está embebido en sistemas que pocos entienden por completo. DOCMAIN lo vuelve explícito.



Figura 1. El conocimiento embebido en el código se vuelve explícito, trazable y gobernado.

Este problema se traduce en costos y riesgos concretos: cambios pequeños con impactos imprevistos, migraciones que se encarecen al mover código sin entender el negocio que sostiene, y un ecosistema DevOps que muestra *qué* cambió y *cuándo*, pero no *qué significa* ese cambio para el negocio.

3. Propuesta de valor

DOCMAIN aporta seis valores que se refuerzan entre sí. Juntos convierten el código de un activo opaco en un activo gobernado y consultable.

Valor	Qué aporta a la organización
Entendimiento de aplicaciones	Identifica qué hacen los sistemas, qué módulos existen, qué reglas implementan y qué procesos soportan.
Fuente de verdad del negocio	Transforma código, documentos, APIs, tablas, jobs y evidencias en conocimiento consultable, trazable y gobernado.
Cambios con menor riesgo	Revela los componentes, procesos, APIs, tablas y reglas afectadas antes de modificar una funcionalidad.
Migración y modernización	Permite decidir qué mantener, encapsular, reescribir, migrar, retirar o investigar — con criterio, no por intuición.
Visualización del negocio	Conecta procesos con aplicaciones, reglas, datos, interfaces e infraestructura en una vista única.
Complemento a DevOps	Agrega una capa semántica, documental y de impacto sobre Git, pipelines, tickets y herramientas de desarrollo.

Frase clave

El código no se usa solo para migrar. Se usa para entender el negocio.

4. El código como fuente de verdad

En muchas empresas, el código es la expresión más precisa de cómo opera el negocio. Una política comercial, una regla de cálculo, una validación, una excepción operativa o un proceso crítico puede estar implementado en **COBOL, Java, .NET, PL/SQL, Oracle Forms, PowerBuilder, Python, scripts batch, APIs o procedimientos almacenados**. Esa lógica es la verdad operativa de la organización, aunque no esté documentada en ningún otro lugar.

DOCMAIN parte de esa premisa: si el negocio vive en el código, entonces el código debe poder leerse, consultarse y gobernarse como conocimiento de negocio, no solo como texto técnico. La plataforma extrae ese significado y lo conecta con los procesos, datos y reglas que representa.

5. Arquitectura técnica por capas

La plataforma se organiza en seis capas que llevan los artefactos de software, paso a paso, desde su forma cruda hasta un conocimiento validado y gobernado. Cada capa produce un resultado concreto que alimenta a la siguiente.



BIZNET DOCMAIN Code Intelligence · Arquitectura técnica de referencia

Figura 2. Arquitectura de referencia. El conocimiento fluye de la ingesta al gobierno, con la capa agentic como motor central.

Capa	Función técnica	Resultado
Ingesta	Conecta repos Git/SVN, bases de datos, APIs, jobs, pipelines, tickets, documentación y evidencias.	Fuentes registradas y artefactos versionados, listos para análisis.
Parsing	Extrae programas, clases, métodos, copybooks, JCL, SQL, APIs, variables y configuraciones.	Modelo normalizado de activos técnicos y funcionales.
Base vectorial	Convierte código y documentos en embeddings para búsqueda semántica y RAG técnico.	Consultas por significado sobre código y documentación.
Grafo de conocimiento	Representa relaciones explícitas entre programas, tablas, APIs, jobs, procesos y reglas.	Mapa de dependencias, análisis de impacto y linaje.

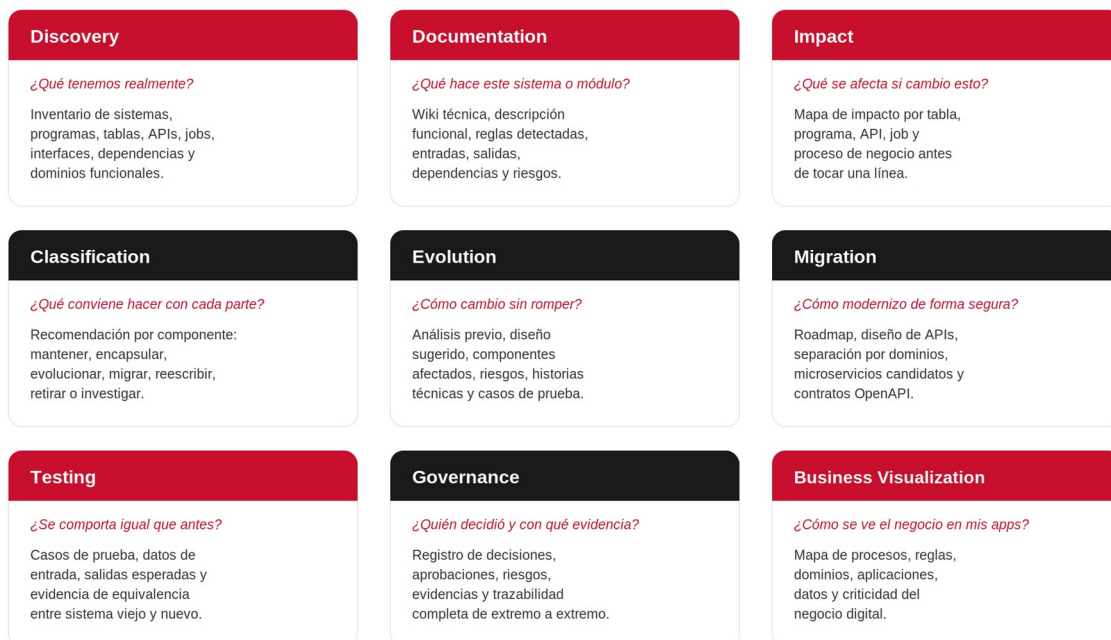
Capa	Función técnica	Resultado
Wiki técnica	Genera fichas de sistemas y módulos con propósito, entradas, salidas, reglas, riesgos y evidencias.	Documentación viva revisable por humanos.
Capa agentic	Orquesta agentes especializados sobre código, grafo, vectores, documentos y sistemas externos.	Discovery, documentación, impacto, evolución, migración, testing y gobierno.
Gobierno	Registra decisiones, aprobaciones, riesgos, versiones, cambios, pruebas y trazabilidad.	Control formal, auditoría y confianza en el conocimiento.

6. Capa agentic: agentes especializados

Los agentes son el motor de la plataforma. No operan como cajas negras: **toda conclusión relevante queda respaldada por evidencia** — archivo fuente, fragmento de código, relación de grafo, documento, ticket, prueba o aprobación humana. Cada agente responde una pregunta concreta del negocio.

La capa agentic: 9 agentes especializados

Cada agente responde una pregunta del negocio. Ninguno opera como caja negra: toda conclusión queda respaldada por evidencia.



Rojo = entendimiento y acción · Negro = clasificación, evolución y gobierno

Figura 3. Los nueve agentes especializados y la pregunta de negocio que responde cada uno.

Agente	Pregunta que responde	Salida principal
Discovery	¿Qué tenemos realmente?	Inventario de sistemas, programas, tablas, APIs, jobs, interfaces, dependencias y dominios.
Documentation	¿Qué hace este sistema o módulo?	Wiki técnica, descripción funcional, reglas, entradas, salidas, dependencias y riesgos.
Impact	¿Qué se afecta si lo cambio o migro?	Mapa de impacto por tabla, programa, API, job y proceso de negocio.
Classification	¿Qué conviene hacer con cada parte?	Recomendación: mantener, encapsular, evolucionar, migrar, reescribir, retirar o investigar.
Evolution	¿Cómo cambio sin romper el sistema?	Análisis previo, diseño sugerido, componentes afectados, riesgos, historias y casos de prueba.

Agente	Pregunta que responde	Salida principal
Migration	¿Cómo modernizo de forma segura?	Roadmap, diseño de APIs, separación por dominios, microservicios candidatos y contratos OpenAPI.
Testing	¿Se comporta igual que antes?	Casos de prueba, datos de entrada, salidas esperadas y evidencia de equivalencia.
Governance	¿Quién decidió y con qué evidencia?	Registro de decisiones, aprobaciones, riesgos, evidencias y trazabilidad.
Business Visualization	¿Cómo se ve el negocio en mis apps?	Mapa de procesos, reglas, dominios, aplicaciones, datos y criticidad.

7. Casos de uso principales

7.1 Migración de sistemas legados

DOCMAIN permite entender el sistema antes de migrarlo. La migración deja de ser un salto a ciegas y se convierte en un proceso progresivo, gobernado y basado en evidencia.

- Inventario de aplicaciones y módulos
- Mapa de dependencias entre componentes
- Reglas de negocio detectadas
- Clasificación de componentes por estrategia
- Candidatos a encapsular, reescribir, migrar o retirar
- Roadmap de migración
- Casos de prueba de equivalencia

7.2 Cambios evolutivos

Antes de modificar una funcionalidad, DOCMAIN analiza impacto y dependencias. El cambio se ejecuta con más contexto, menos riesgo y mejor trazabilidad.

- Componentes afectados
- Reglas de negocio relacionadas
- APIs y tablas impactadas
- Riesgos de regresión identificados
- Casos de prueba recomendados
- Documentación actualizada y evidencia para aprobación

7.3 Visualización del negocio digital

DOCMAIN muestra cómo el negocio está implementado en los sistemas, conectando procesos, reglas, datos, aplicaciones y dominios funcionales en una vista única.

- Mapa de procesos
- Mapa de aplicaciones
- Mapa de reglas de negocio
- Relación entre módulos y capacidades de negocio
- Vista ejecutiva de criticidad y dependencias

7.4 Auditoría, cumplimiento y continuidad

DOCMAIN justifica cambios, controles y decisiones con evidencia, y reduce la dependencia de personas clave al transformar conocimiento disperso en una base consultable y gobernada.

- Evidencias de pruebas
- Trazabilidad de decisiones
- Registro de aprobaciones
- Relación entre normativa, proceso, sistema y código
- Wiki técnica y funcional para operación y soporte

8. Flujo técnico de trabajo

De conectar las fuentes a obtener una fuente de verdad validada y trazable, el trabajo se estructura en siete pasos. Una primera implementación puede ejecutarse sobre un único dominio o aplicación crítica, sin migrar nada al inicio.

Flujo técnico de trabajo

De conectar las fuentes a una fuente de verdad validada y trazable — en siete pasos.



Una primera implementación puede ejecutarse sobre un único dominio o aplicación crítica, sin migrar nada al inicio.

Figura 4. Flujo técnico de extremo a extremo, del conector a la fuente de verdad gobernada.

Paso	Descripción técnica	Salida
1. Conectar fuentes	Repositorios, bases de datos, documentos, tickets y pipelines.	Fuentes registradas y versionadas.
2. Detectar artefactos	COBOL, copybooks, JCL, Java, .NET, SQL, scripts, YAML, OpenAPI, documentos.	Inventario inicial de artefactos.
3. Extraer entidades	Llamadas, imports, COPY, EXEC SQL, tablas, endpoints, jobs y variables.	Entidades técnicas normalizadas.
4. Crear chunks semánticos	Programa, sección, función, query, procedimiento, regla o documento.	Fragmentsos listos para embeddings y RAG.
5. Construir grafo	Programa usa tabla, job ejecuta programa, API expone proceso, regla pertenece a dominio.	Grafo de dependencias y conocimiento.
6. Generar documentación	Los agentes producen fichas, resúmenes, reglas, riesgos y salidas.	Wiki técnica y funcional.
7. Validar y gobernar	Los usuarios revisan, aprueban, corrigen o enriquecen la información.	Fuente de verdad validada y trazable.

9. Relación con DevOps, Git y CI/CD

DOCMAIN **no reemplaza** Git, DevOps ni CI/CD. Git responde qué cambió, quién y cuándo. DevOps responde cómo se construye, prueba y despliega el software. DOCMAIN responde la pregunta que ninguno responde: **qué significa ese software para el negocio**, qué reglas implementa, qué depende de qué y qué impacto tendría modificarlo.

La capa que falta sobre el ciclo de vida del software

DOCMAIN no reemplaza Git ni DevOps. Responde la pregunta que ninguno responde: ¿qué significa este software para el negocio?

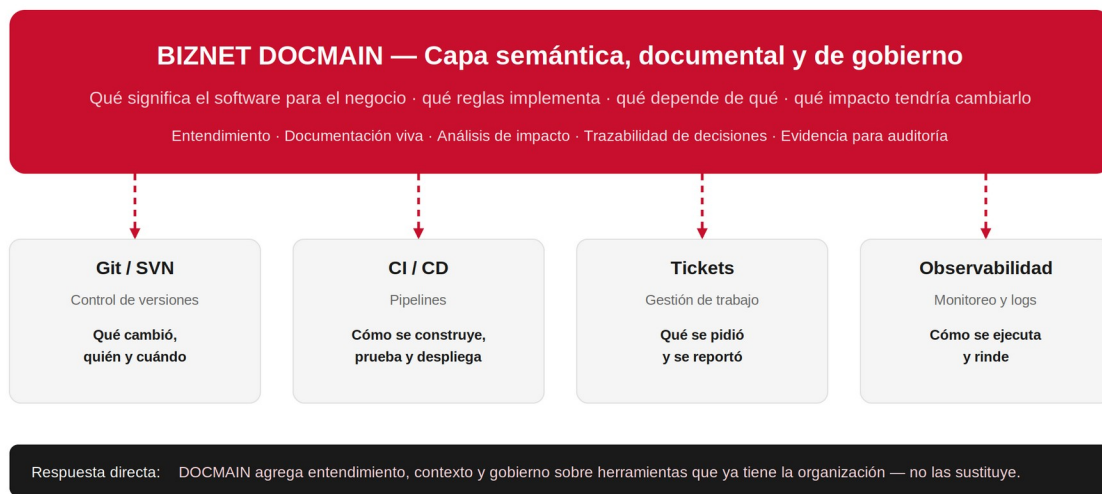


Figura 5. DOCMAIN aporta la capa de significado de negocio sobre herramientas que la organización ya tiene.

Elemento	Rol principal	DOCMAIN complementa con
Git / SVN	Control de versiones del código.	Contexto semántico, documentación, relaciones de negocio e impacto.
CI / CD	Compilar, probar y desplegar.	Análisis de impacto previo, evidencias funcionales y trazabilidad.
Tickets	Gestionar requerimientos, incidentes y cambios.	Relación entre historia, regla de negocio, código, prueba y evidencia.
Observabilidad	Monitorear ejecución, logs y rendimiento.	Relación de fallas operacionales con componentes, procesos y reglas.

Respuesta directa

DOCMAIN agrega una capa semántica, documental, relacional y de gobierno sobre todo el ciclo de vida del software.

10. Extensión hacia infraestructura y operaciones

Aunque la entrada comercial principal es el código, una vez entendido el software la plataforma puede extenderse hacia la infraestructura y la operación: servidores, máquinas virtuales, contenedores, Kubernetes, bases de datos, redes, firewalls, balanceadores, monitoreo, logs, tickets, manuales y CMDB.

Visión extendida: una fuente de verdad integral de TI

El código es la entrada. La plataforma se extiende hacia dónde vive y cómo opera esa lógica.



Responde: ¿dónde corre una app? ¿qué pasa si cae un servidor? ¿qué servicios dependen de esta base de datos? ¿qué incidentes se ligan a este componente?

Figura 6. Code Intelligence entiende la lógica del negocio; Infrastructure Intelligence entiende dónde vive y cómo opera.

Esta extensión permite responder preguntas operativas que hoy requieren reunir a varias personas: dónde corre una aplicación, qué infraestructura soporta un proceso, qué ocurre si cae un servidor, qué servicios dependen de una base de datos y qué incidentes se relacionan con un componente.

Visión extendida

Code Intelligence entiende la lógica del negocio. Infrastructure Intelligence entiende dónde vive y cómo opera esa lógica. Juntas construyen una fuente de verdad integral de TI.

11. Entregables de una primera implementación

Una primera implementación se realiza sobre un dominio o aplicación crítica, sin necesidad de migrar nada al inicio. Los entregables tangibles son:

- **Inventario** — de aplicaciones, módulos, tablas, APIs y jobs.
- **Grafo inicial** — de dependencias entre componentes.
- **Base vectorial** — para consultas semánticas sobre código y documentos.
- **Wiki técnica** — generada automáticamente y revisable.
- **Mapa de procesos** — y reglas de negocio detectadas.
- **Informe de impacto** — para los cambios seleccionados.
- **Clasificación** — de componentes por estrategia recomendada.
- **Roadmap preliminar** — de evolución o migración.
- **Casos de prueba base** — para validar equivalencia funcional.
- **Panel de gobierno** — con evidencias y riesgos.

12. Posicionamiento comercial

BIZNET DOCMAIN Code Intelligence se posiciona como la plataforma para entender, documentar, visualizar, evolucionar y modernizar aplicaciones críticas a partir del código y sus dependencias.

Mensaje	Qué significa para la organización
El código es conocimiento del negocio	DOCMAIN convierte ese conocimiento en una base consultable, documentada y trazable.
Antes de migrar, entender	Permite saber qué existe, qué hace, qué depende de qué y qué riesgo tiene cada cambio.
Cambios con evidencia	Habilita evolutivos con análisis de impacto, trazabilidad y validación.
Visualización del negocio digital	Conecta código, procesos, datos y reglas para entender cómo opera la organización.
Complemento a DevOps	No reemplaza pipelines ni Git: agrega entendimiento semántico, documentación viva y gobierno.

13. Conclusión

BIZNET DOCMAIN Code Intelligence transforma el código y los artefactos de software en una fuente de verdad del negocio digital. Su valor no está solo en migrar: está en permitir que la organización entienda sus aplicaciones críticas, visualice sus procesos, analice impactos, gobierne cambios, reduzca riesgos y tome decisiones técnicas y funcionales con evidencia.

La promesa correcta

No prometemos migrar millones de líneas automáticamente. Prometemos entender, documentar y gobernar millones de líneas de código para decidir, con evidencia, qué evolucionar, modernizar, encapsular, migrar, mantener o retirar.



BIZNET · Gestión Documental Cognitiva y Agentic