



BOHIOSYSTEM

• ENTREGABLES Y EJEMPLO DE ADR • BOHIOSYSTEM

Entregables de Arquitectura

Qué recibe exactamente el cliente en un encargo de arquitectura, con un ejemplo real de registro de decisión.

DOCUMENTO

BS-SVC-02B

VERSIÓN

1.0

FECHA

1 de agosto de 2026

PREPARADO POR

BohioSystem

QUÉ SE ENTREGA

“Diseño de arquitectura” significa cosas muy distintas según quién lo venda. Este documento define exactamente qué recibes.

Todos los entregables se escriben en Markdown y se versionan en el repositorio del proyecto, junto al código. Los diagramas se generan desde texto, de modo que se actualizan con el mismo flujo de trabajo que el software y no quedan desfasados.

4

VISTAS DEL MODELO C4

1 ADR

POR DECISIÓN
RELEVANTE

Markdown

FORMATO VERSIONABLE

Texto

DIAGRAMAS COMO CÓDIGO

01

Las cuatro vistas

Usamos el modelo C4 porque separa el nivel de detalle según la audiencia: dirección, equipo de producto y equipo técnico no necesitan el mismo diagrama.

VISTA	RESPONDE A	PARA QUIÉN
Contexto	¿Qué hace el sistema y con quién habla?	Dirección, negocio y cualquier persona nueva en el proyecto.
Contenedores	¿De qué piezas desplegables se compone?	Equipo técnico y responsables de infraestructura.
Componentes	¿Cómo se organiza cada pieza por dentro?	Equipo de desarrollo que va a construirlo.
Código	¿Cómo se implementa un punto concreto?	Solo donde aporta: se genera bajo demanda, no se mantiene.

02

Paquete completo de entregables

Documento de arquitectura

- Contexto de negocio y restricciones.
- Vistas C4 de contexto, contenedores y componentes.
- Atributos de calidad con sus objetivos.
- Riesgos identificados y su mitigación.

Registro de decisiones

- Un archivo por decisión, numerado.
- Contexto, opciones, decisión y consecuencias.
- Estado: propuesta, aceptada, sustituida.
- Enlace a la decisión que la reemplaza.

Modelo de datos

- Diagrama entidad-relación.
- Diccionario de datos por entidad.
- Fuentes de verdad por dominio.
- Política de retención y archivado.

Plan de implementación

- Fases con sus dependencias.
- Criterios de entrada y salida por fase.
- Estimación de esfuerzo.
- Riesgos y plan de contingencia.

03

Qué es un ADR y por qué importa

Un Architecture Decision Record es un archivo corto que registra una decisión técnica y, sobre todo, por qué se tomó.

El valor de un ADR no está en la decisión, está en el contexto. Dentro de dos años, cuando alguien pregunte “¿por qué usamos esto y no aquello?”, la respuesta estará escrita junto al código, con las alternativas que se evaluaron y lo que se aceptó a cambio. Sin ese registro, el equipo vuelve a discutir lo mismo o —peor— revierte una decisión sin entender qué problema resolvía.

REGLA PRÁCTICA

Se escribe un ADR cuando la decisión es **cara de revertir**: elección de base de datos, límite entre servicios, protocolo de integración, estrategia de autenticación, modelo de despliegue. No se escribe para elegir una librería de fechas.

04

Ejemplo real de ADR

Este es el formato exacto que entregamos, con un caso representativo.

CAMPO	CONTENIDO
Identificador	ADR-014
Título	Monolito modular en lugar de microservicios para la fase inicial
Estado	Aceptada · sustituye a ADR-006
Contexto	La plataforma tendrá cuatro dominios funcionales. El equipo es de cinco personas y no cuenta hoy con operación 24/7 ni con experiencia previa en sistemas distribuidos. Se espera crecimiento, pero la carga del primer año es moderada y los límites entre dominios todavía no están estabilizados.
Opciones	A. Microservicios desde el inicio. B. Monolito modular con límites internos explícitos. C. Monolito sin separación de módulos.
Decisión	Opción B. Un despliegue único, con cuatro módulos de límites explícitos, comunicación interna mediante interfaces y una base de datos por esquema separado.

CAMPO	CONTENIDO
Motivo	Los límites entre dominios aún se están descubriendo, y equivocarse en el corte de un microservicio es mucho más caro de corregir que mover código entre módulos. La opción A añade complejidad operativa que el equipo hoy no puede sostener; la C impediría separar más adelante.
Consecuencias	Positivas: despliegue simple, depuración local completa, transacciones sin coordinación distribuida. Negativas: no se puede escalar un módulo por separado, y un fallo grave afecta a todo el proceso. Se acepta a cambio de poder extraer un módulo a servicio independiente cuando su carga lo justifique, sin reescribir la lógica.
Revisión	Se reevalúa si un módulo supera el 60% del consumo total o si el equipo crece por encima de doce personas.

05

Atributos de calidad

Toda arquitectura sacrifica algo. Lo que la hace defendible es haber elegido qué sacrificar, y haberlo escrito.

ATRIBUTO	PREGUNTA QUE RESPONDE	CÓMO SE FIJA EL OBJETIVO
Disponibilidad	¿Cuánto tiempo puede estar caído?	Porcentaje mensual y ventana máxima de indisponibilidad.
Rendimiento	¿Cuánto puede tardar una operación?	Tiempo de respuesta en el percentil 95 bajo carga definida.
Escalabilidad	¿Hasta dónde crece sin rediseñar?	Carga objetivo a 12 y 36 meses.
Seguridad	¿Qué protege y de quién?	Modelo de amenazas y controles por capa.
Mantenibilidad	¿Cuánto cuesta cambiarlo?	Tiempo de incorporación de un desarrollador nuevo.
Recuperabilidad	¿Cuánto se tarda en volver y cuánto dato se pierde?	Objetivos de tiempo y punto de recuperación.

EL COMPROMISO CENTRAL

Un sistema que cumple todos los atributos al máximo no existe. Nuestro trabajo es hacer explícito el intercambio y que sea el negocio, no el equipo técnico en solitario, quien lo apruebe.

Hablemos

SIGUIENTE PASO

Escríbenos y agendamos una sesión de trabajo sin costo para revisar tu caso concreto.

CONTACTO

contacto@bohiosystem.dev

WEB

bohiosystem.dev

DOCUMENTO

BS-SVC-02B · v1.0