

Acta de Reunion #1

KairosMix1

Fecha	Versión	Autor	Verificado dep. calidad.
10/12/2025	1	Matías Lugmaña, Camilo Orrico, Denise Rea, Julio Viche	Mgt. Jenny Alexandra Ruiz Robalino

ACTA DE REUNIÓN DE PROYECTO - KAIROZ MIX

Código de Acta: KAIROZ-ACT-001

Fecha: 20 de Noviembre, 2025

Hora: 17:00 PM - 18:30 PM

Proyecto: Sistema de Gestión Kairoz Mix

Módulo: Integración de Gestión de Clientes y Arquitectura de Capas

1. ASISTENTES

Por parte del Cliente / Interesados:

- Vinicio Narvez

Por parte del Equipo de Desarrollo (Kairoz Mix Dev Team):

- Matias Lugmaña
- Denis Rea
- Julio Viche
- Camilo Orrico

1.1 Evidencia Reunión

url: <https://youtu.be/U1RIC9GvFNU>

2. AGENDA DEL DÍA

1. Presentación del flujo de arquitectura técnica para el módulo de Clientes.
2. Validación de la integración entre ClientApp y el sistema de pedidos.
3. Revisión de la estructura de capas (Presentación, API, Acceso a Datos, Persistencia).
4. Acuerdos de desarrollo.

3. DESARROLLO DE LA REUNIÓN

3.1. Presentación de la Arquitectura Técnica

El equipo técnico expuso el diagrama de arquitectura actualizado para soportar el módulo de clientes, detallando las 4 capas principales definidas en el proyecto Kairoz Mix:

- Capa de Presentación (Frontend - React + Vite):
Se definió que el punto de entrada será el componente ClientApp, el cual reutilizará la lógica de UI existente (como ProductModal y ProductTable) para mantener consistencia visual. La comunicación con el backend se realizará mediante clientService (API Client).
- Capa de API (Backend - Express Router):
Se establecieron las rutas en clientRoutes, las cuales orquestarán las peticiones HTTP entrantes y las derivarán a los controladores correspondientes, separando claramente las rutas de clientes de las de productos (productRoutes).
- Capa de Controladores:
Se validó la estructura interna de clientController, confirmando los métodos principales necesarios para el MVP:
 - registerClient: Para el alta de nuevos usuarios.
 - getClient: Para la recuperación de datos de perfil.
 - viewClientOrders: Para la trazabilidad histórica de compras.
- Capa de Acceso a Datos y Persistencia:
Se explicó el uso del patrón Repository. Aunque el diagrama muestra MongoProductRepository, se acordó implementar un MongoClientRepository homólogo que implemente la interfaz correspondiente para interactuar con la clase Database (Singleton) y finalmente con MongoDB.

3.2. Revisión Funcional (Historias de Usuario)

Se discutió cómo esta arquitectura soporta las necesidades del negocio:

- **Integridad:** La separación en capas asegura que si se cambia la base de datos a futuro, el Frontend (ClientApp) no se verá afectado.
- **Escalabilidad:** El uso de interfaces en los repositorios permite añadir nuevas reglas de negocio para clientes VIP sin romper el código existente.

3.3. Retroalimentación y Aprobación

El Cliente validó que los campos considerados en el clientController (datos personales + historial) cubren los requisitos actuales de facturación y logística.

Decisión: Se aprueba la arquitectura técnica propuesta para proceder con el código.

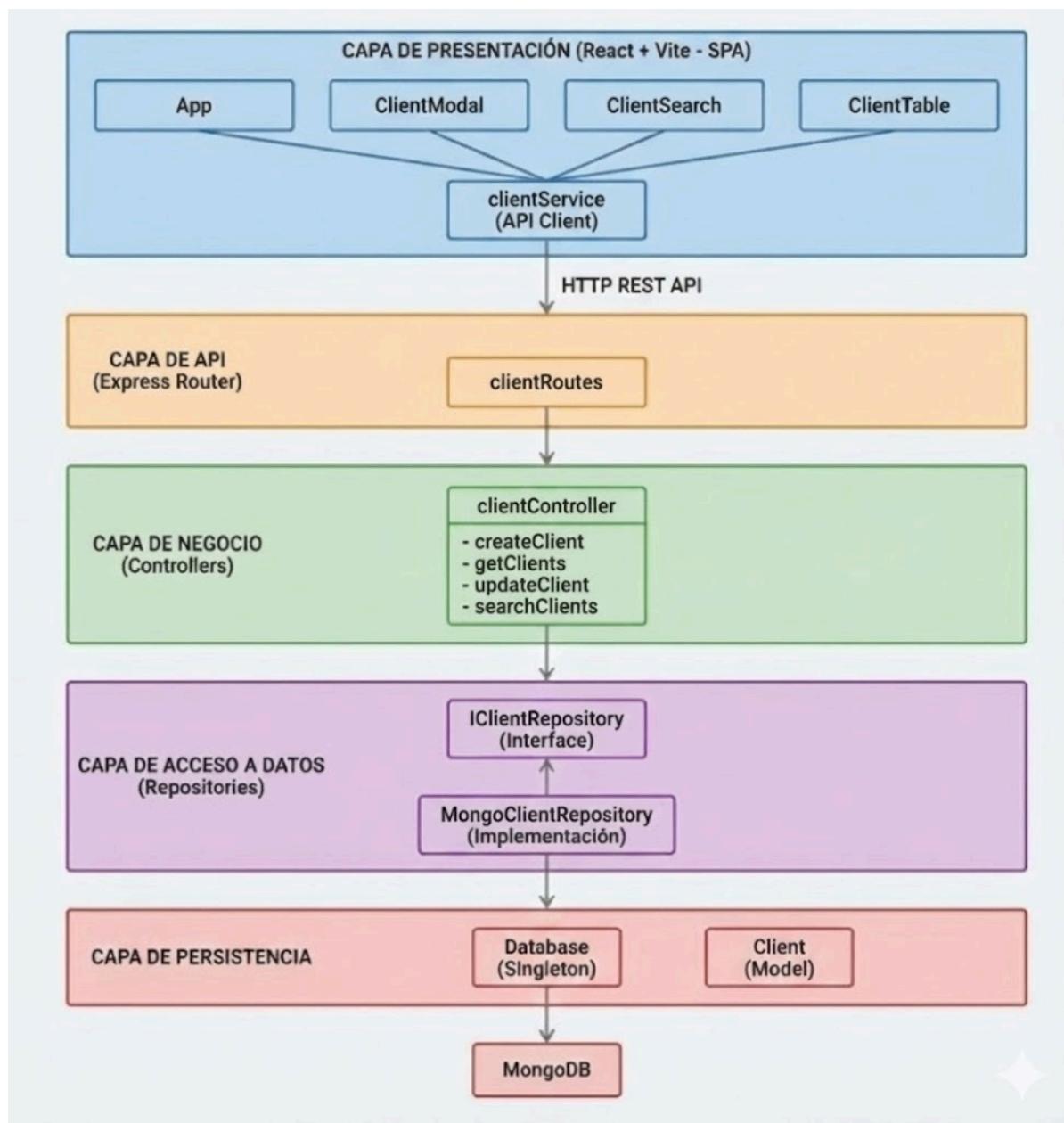
4. ACUERDOS Y COMPROMISOS

ID	Acuerdo / Tarea	Responsable	Fecha Límite	Estado

01	Implementar <code>clientRoutes</code> y conectar con <code>clientController</code> en Express.	Backend Dev	18/12/2025	Pendiente
02	Crear la interfaz <code>IClientRepository</code> y su implementación en Mongo.	Backend Dev	19/12/2025	Pendiente
03	Desarrollar vista <code>ClientApp</code> en React consumiendo <code>clientService</code> .	Frontend Dev	20/12/2025	Pendiente
04	Pruebas de integración (Endpoint <code>registerClient</code>).	QA / Dev	21/12/2025	Pendiente

5. EVIDENCIA DE LA REUNIÓN

Se adjunta el diagrama de arquitectura presentado (v.2.0 - Integración Clientes) como anexo técnico a esta acta.



Firmas de conformidad:

[Nombre del Cliente]

Aprobación del Proyecto

[Tu Nombre]

Líder de Desarrollo - Kairoz Mix