

**MISO**

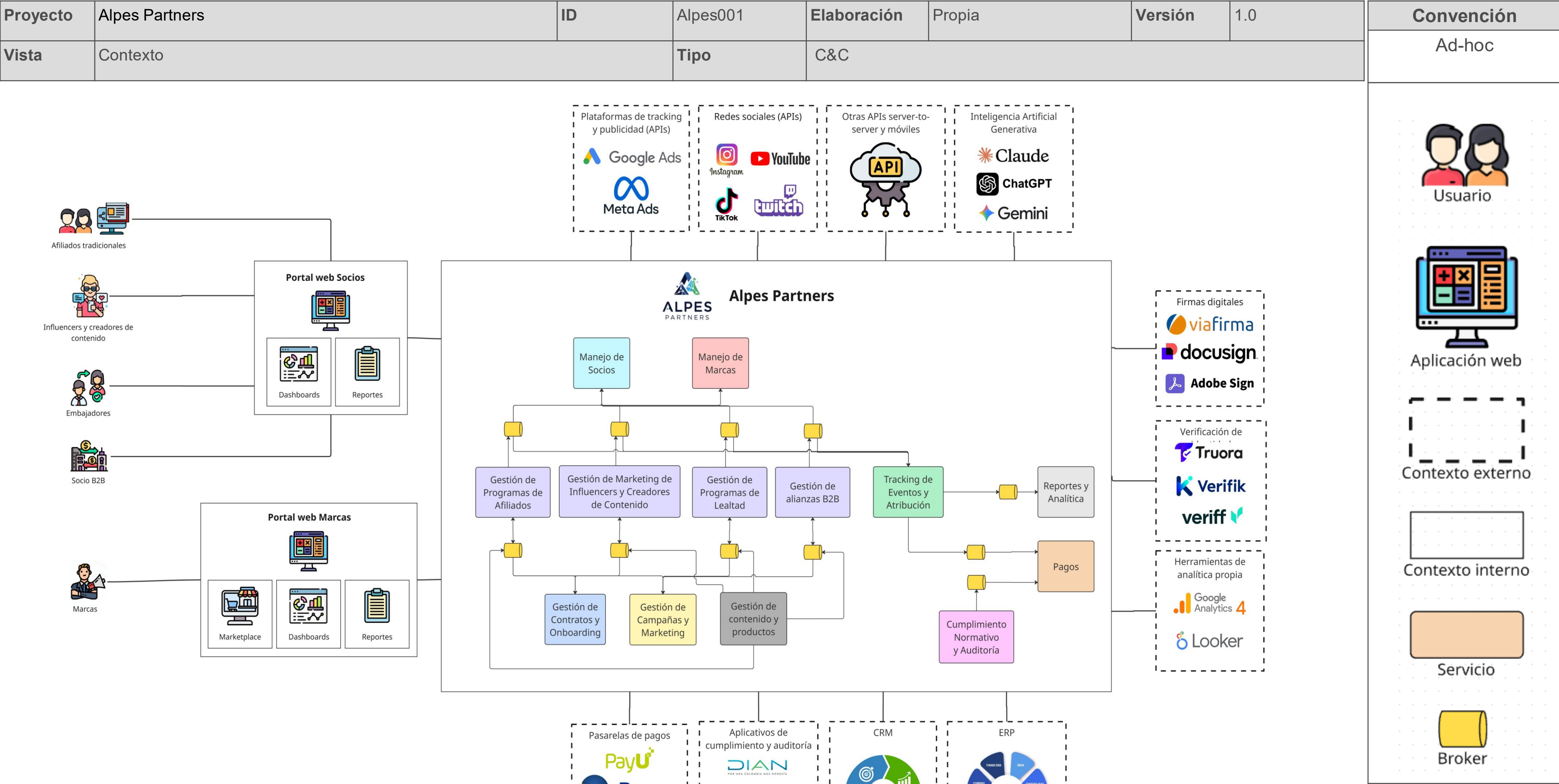
Maestría en Ingeniería de Software

## Entrega 2: Reactive Builders

# Requerimientos de calidad

Atributo de calidad	Prioridad (L/M/H)	Justificación
Modificabilidad	H	<p>La arquitectura monolítica actual dificulta la modificabilidad y genera un alto nivel de acoplamiento, lo que provoca que cambios en un componente puedan afectar a todo el sistema, comprometiendo su estabilidad y disponibilidad. Además, el desarrollo y pruebas de nuevas funcionalidades se vuelve cada vez más lento y costoso, limitando la capacidad de adaptación del producto.</p> <p>De igual forma, los tiempos de prueba se ven significativamente impactados, pues cada modificación requiere validar el sistema completo, retrasando la entrega de valor y aumentando los riesgos de regresión.</p> <p>Por estas razones, resulta necesario avanzar hacia una arquitectura más flexible y escalable, que fomente la autonomía de los equipos, reduzca el acoplamiento y acelere tanto el desarrollo como la validación de cambios.</p>
Escalabilidad	H	<p>La escalabilidad se menciona explícitamente como una consecuencia directa de lo que implica un IPO.</p> <p>El fortalecimiento de la escalabilidad no solo aporta beneficios propios, sino que también puede impactar de forma positiva en otros atributos de calidad, como la disponibilidad y el desempeño, ambos siendo mencionados dentro de los SLA.</p> <p>También, la tabla muestra que existe un gran número de transacciones, por lo que tener un sistema que pueda soportarlas y escalar sería recomendable.</p> <p>Finalmente, la escalabilidad resulta un requisito esencial dentro de una arquitectura orientada a eventos, dado que este estilo arquitectónico depende de la capacidad de manejar altos volúmenes de información y cargas variables de manera eficiente.</p>
Disponibilidad	H	<p>En Alpes Partners la disponibilidad es crítica porque la plataforma conecta marcas y socios en procesos de asociaciones, tracking de eventos, contratos y pagos que no pueden detenerse, al menos desde el punto de vista del usuario.</p> <p>Al ser una plataforma global que opera 24/7 en todo momento habrá operaciones críticas. Cualquier interrupción en servicios como asociaciones, tracking de eventos, contratos o pagos impacta directamente en la confianza de socios y marcas, y puede representar pérdidas económicas significativas.</p> <p>Por ello, el sistema debe asegurar continuidad aún bajo fallos parciales, permitiendo que los usuarios perciban siempre un servicio estable y confiable.</p>

# Puntos de vista





# Decisiones de diseño y descripción

- El diagrama ilustra la arquitectura general del sistema de Alpes Partners, la cual se basa en un enfoque de microservicios orientados a eventos. En total, se han implementado 8 microservicios, que en conjunto soportan y habilitan las principales capacidades de negocio. Los servicios finales son: Manejo de marcas y socios, gestión de asociaciones estratégicas, gestión de campañas y ofertas, gestión de contratos, tracking de eventos y atribución, reportes y analítica, pagos, cumplimiento normativo y auditoría.
  - La adopción de este estilo arquitectural busca incrementar la velocidad y la **modificabilidad** del sistema. Al separar las funcionalidades en servicios más pequeños y autónomos, se reduce la dependencia entre equipos y se facilita la evolución continua del producto. De igual forma, la arquitectura de microservicios se eligió para mejorar la **escalabilidad**, un atributo especialmente importante debido al crecimiento en clientes debido al IPO.
  - La fragmentación del monolito en componentes independientes permite aislar la carga de trabajo, evitando los problemas de vecinos ruidosos y habilitando un mejor rendimiento bajo condiciones de alta demanda. Finalmente, el diseño también prioriza el **desempeño** del sistema, asegurando el cumplimiento de los SLA establecidos, como la activación de campañas en menos de 24 horas y la ejecución de pagos a afiliados en un máximo de 48 horas. Para alcanzar estos objetivos, se han separado los procesos críticos mediante brokers y se ha dotado a cada microservicio de mecanismos de escalado independiente, con el fin de eliminar cuellos de botella y mitigar degradaciones de rendimiento provocadas por cargas desbalanceadas.
- **Puntos de sensibilidad:**
- El cumplimiento de los SLA clave (24h en campañas y 48h en pagos) está estrechamente ligado al rendimiento del sistema. Cualquier incremento en la latencia o disminución en el throughput puede comprometer la satisfacción del cliente y generar incumplimientos contractuales.
  - Una arquitectura distribuida incrementa el riesgo de vecinos ruidosos si no existe un aislamiento adecuado de recursos. Una sobrecarga en un microservicio puede repercutir en otros, afectando la estabilidad general.
  - La observabilidad se convierte en un factor esencial: sin monitoreo en tiempo real de métricas como latencia, tasa de procesamiento y errores, resulta complejo anticipar fallas, localizar cuellos de botella y garantizar la continuidad del servicio.
- **Puntos de tradeoff:** -
- Aunque la separación de servicios facilita la escalabilidad y disminuye la dependencia del monolito, también introduce una mayor complejidad operacional, ya que se requiere una infraestructura avanzada de orquestación, trazabilidad distribuida y gestión de fallas. La arquitectura implica aceptar consistencia eventual en lugar de consistencia inmediata, lo cual puede impactar en la experiencia del usuario en determinados escenarios. El rendimiento del sistema ya no depende exclusivamente de la capacidad de cómputo, sino también de la infraestructura de red y de la eficiencia de múltiples interacciones asíncronas, aumentando los puntos de vulnerabilidad en comparación con un monolito donde la comunicación era interna y directa.

# Puntos de vista

Proyecto	Alpes Partners	ID	Alpes002	Elaboración	Propia	Versión	1.0	Convención
Vista	Funcional	Tipo		Módulo				UML

# Decisiones de diseño y descripción

## - **Análisis de Diseño y Descripción**

- Como se puede apreciar en la vista de módulos, se ha definida una arquitectura basada en el patrón de modelo de dominio e inversión de dependencias, siguiendo los lineamientos de DDD.
- Cada módulo cuenta con tres capas separadas: Aplicación, Dominio e Infraestructura. El dominio se mantiene aislado de dependencias externas.
- Esta separación en capas nos asegura que las decisiones técnicas de tipo bases de datos, tecnologías, infraestructura, etc. puedan ser modificadas con un impacto menor sobre el núcleo del negocio.

## - **Puntos de sensibilidad:**

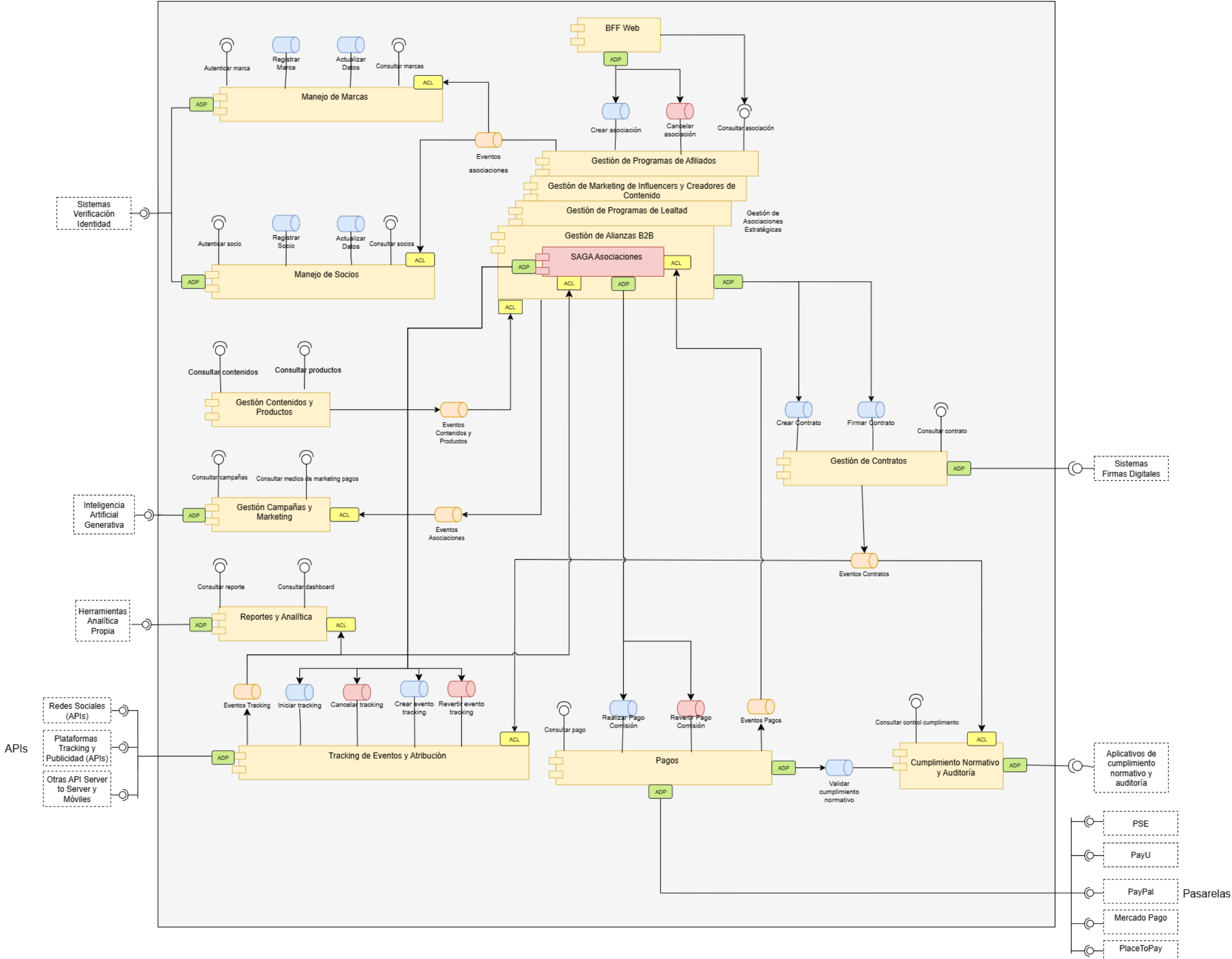
- El respeto a las interfaces y segregación de responsabilidades es clave para la correcta aplicación de DDD e inversión de dependencias.
- Sin lo anterior, el modelo de dominio podría verse afectado de detalles técnicos, y eso afectaría negativamente la modificabilidad y extensibilidad del sistema.
- Es crítico asegurar que se use de forma sostenida genéricos e interfaces para así mantener desacoplados los módulos.

## - **Puntos de tradeoff: -**

- El uso de múltiples capas y módulos incrementa la dificultad de comprensión del sistema: navegar por el código y hacer diagnósticos puede requerir mayores esfuerzos que bajo un diseño simple.
- El desempeño puede verse afectado al agregar más capas y módulos, aunque el impacto es menor comparado con el beneficio en modificabilidad, mantenibilidad y flexibilidad.

# Puntos de vista

Proyecto	Alpes Partners	ID	Alpes 001	Elaboración	Propia	Versión	1.0	Convención
Vista	Funcional			Tipo	C&C			Ad hoc





# Decisiones de diseño y descripción

## - **Análisis de Diseño y Descripción**

- En nuestra Vista de Funcional tipo C&C se ha definió una Arquitectura de MicroServicios orientada por Eventos. El diseño ha sido desarrollando usando la filosofía de Diseño Guiado por Dominio; nos hemos centrado en los subdominios principales que dan ventaja competitiva al negocio y son la piedra angular del sistema: Gestión de Asociaciones Estratégicas que permiten asociar las marcas y los socios basados en rendimiento y analítica, Tracking de Eventos y Atribución para dar registro y seguimiento a las conversiones, la Gestión automatizada de Contratos y los Pagos de las comisiones cumpliendo con la normativa en diferentes regiones.
- Como se puede ver en la Vista, se ha optado por una comunicación guiada por eventos, haciendo de los servicios desacoplados entre sí para aislamiento de fallos mediante tópicos de comandos y eventos. También se han implementado patrones de Adaptador y ACLs para evitar contaminar los dominios de los servicios.

## - **Puntos de sensibilidad:**

- La arquitectura favorece la facilidad de modificación, dado que los servicios están desacoplados, permitiendo ser asignados a diferentes equipos alrededor del mundo para implementar nuevas funcionalidades y mejorar el Time To Market. Los microservicios al estar aislados permitirán crecer dinámicamente de manera horizontal para responder a las necesidades de escalabilidad. El rendimiento se beneficia de la posibilidad de ejecutar procesos en paralelo y de manera asíncrona.
- Los brokers de Eventos concentran puntos de fallo por lo cual es importante contar con alta disponibilidad. Afortunadamente, hoy por hoy, la tecnología Cloud ofrece alta disponibilidad en estos componentes, y por ello es importante tener este requerimiento en mente en el momento de seleccionar los componentes y la nube para desplegar.

## - **Puntos de tradeoff:**

- La Arquitectura seleccionada, al ser distribuida, brinda retos al equipo de DevOps para mantener la Observabilidad y encontrar fallos. Es de vital importancia que los microservicios también implementen adecuados Logs para ayudar a encontrar fácilmente errores en el sistema.
- Por otro lado, en el momento de despliegue tenemos que seleccionar componentes de nube en el mismo Cluster de Red para que las comunicaciones de red no comprometan el Desempeño, o usar otras alternativas de implementación con el mismo propósito.



# Puntos de vista

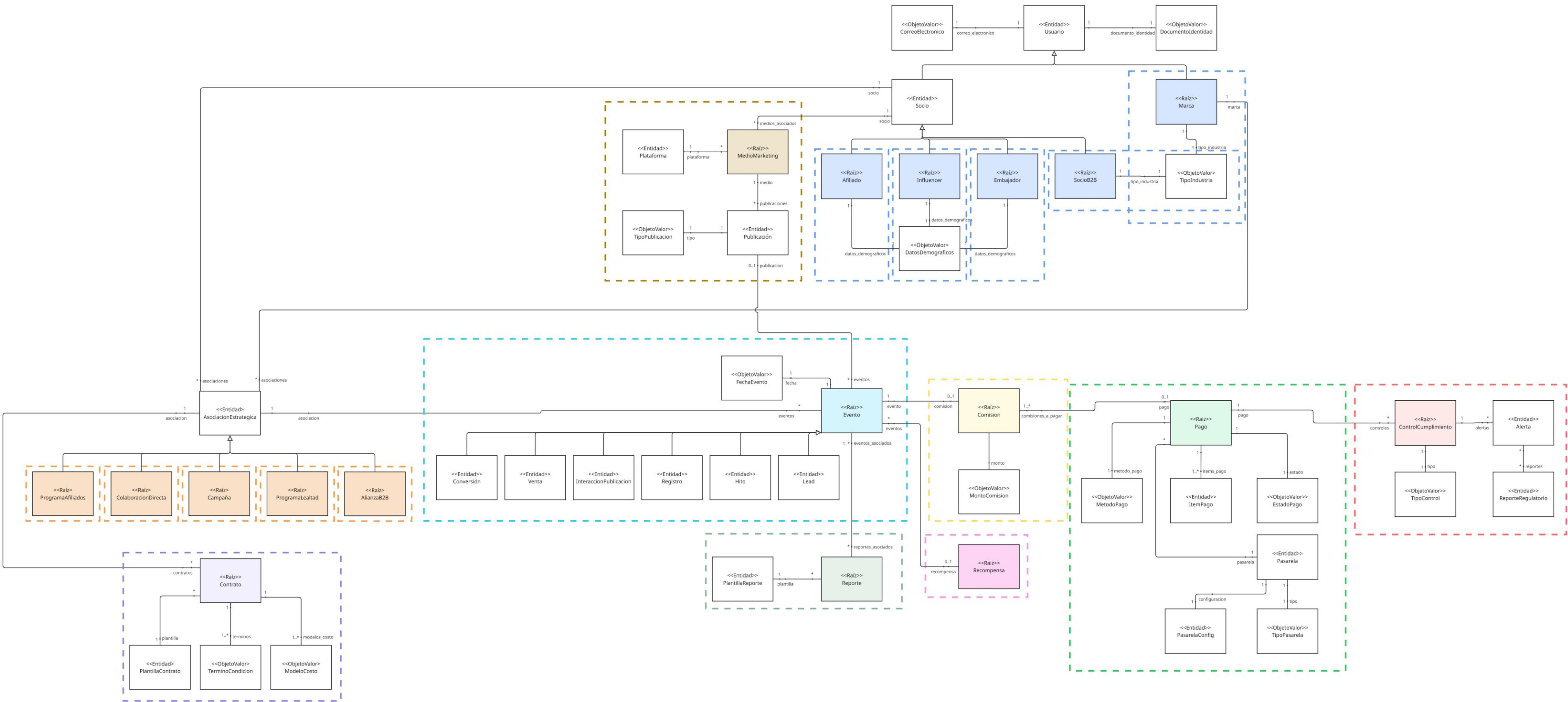
Proyecto	Alpes Partners	ID		Elaboración	Propia	Versión	1.0
Vista	Información		Tipo				

Convención

UML

Agregación

Color: Asociaciones de la misma entidad madre por herencia



# Decisiones de diseño y descripción

## - **Análisis de Diseño y Descripción**

- El diagrama de dominio presentado refleja una visión integral de Alpes Partners, mostrando las principales entidades, objetos de valor y agregaciones que soportan el modelo de partnerships. Se utilizaron herencias y objetos valor. Podemos destacar las siguientes agregaciones principales:

- **5 agregaciones de Asociación Estratégica:** núcleo que articula programas, campañas, colaboraciones directas y actividades de afiliación.

Representa la columna vertebral de la plataforma.

- **5 agregaciones de Usuario:** describe afiliados, influencers, embajadores y socios B2B, incluyendo datos demográficos y empresariales

- **MedioMarketing:** gestiona las publicaciones y plataformas usadas por los socios.

- **Eventos:** clave para la arquitectura orientada a eventos; centraliza conversiones, registros, leads, clicks, entre otros, habilitando el tracking y atribución en tiempo real.

- **Contrato:** abstrae la gestión de términos, condiciones y plantillas, apoyando la flexibilidad legal necesaria en un sistema global.

- **Pago:** cubre el pago de comisiones a través de pasarelas de pago e integración con sistemas de compliance.

## - **Puntos de sensibilidad:**

- La agregación de Eventos será crítica en el modelo, pues de su correcto diseño depende el cumplimiento de SLA como activación de campañas en menos de 24 horas y pagos en 48 horas.

- La interacción entre Contratos y Cumplimiento Normativo es clave para asegurar la escalabilidad internacional y cumplir con normativas diversas (ej. antifraude, auditorías, fiscalidad).

## - **Puntos de tradeoff:**

- La elección de múltiples agregaciones favorece el desacoplamiento y modificabilidad, pero introduce mayor complejidad en la orquestación de eventos y consistencia eventual entre ellas.

- La clara separación entre Pagos y Cumplimiento Normativo es acertada para reducir riesgos regulatorios, pero exige mecanismos de integración robustos y observabilidad en tiempo real.

- El modelo facilita la migración hacia microservicios, pero implica aceptar mayores retos en trazabilidad distribuida y monitoreo, como se discutió en las decisiones de diseño.