**1. Arquitectura Monolítica**

* Todo el sistema está en una sola aplicación.
* Pros: fácil de desarrollar y desplegar al inicio.
* Contras: difícil de escalar y mantener a medida que crecen funcionalidades del CRM (gestión de clientes, ventas, marketing, soporte, etc.).

**2. Arquitectura en Capas (N-Tier)**

* Divide el sistema en capas: presentación, lógica de negocio, acceso a datos, base de datos.
* Pros: ordenado, claro, facilita mantenimiento.
* Contras: puede volverse rígido si el CRM crece mucho, y los cambios en una capa pueden afectar otras.

**3. Arquitectura Orientada a Servicios (SOA)**

* El sistema se divide en **servicios independientes**, que se comunican mediante protocolos estándar (ej. SOAP, REST).
* Pros: integración con otros sistemas empresariales (ERP, facturación, BI).
* Contras: mayor complejidad en la gestión de servicios y comunicación.

**4. Arquitectura de Microservicios**

* Se fragmenta el CRM en **módulos independientes** (usuarios, ventas, marketing, informes), cada uno con su propia lógica y base de datos.
* Pros: escalabilidad, despliegue independiente, se adapta muy bien a CRMs modernos en la nube.
* Contras: requiere mayor experiencia técnica y robustez en la comunicación (API Gateway, balanceadores, etc.).

**5. Arquitectura Basada en Eventos**

* Los componentes reaccionan a **eventos** (ej. “nuevo cliente creado”, “venta cerrada”), usando colas de mensajes (Kafka, RabbitMQ).
* Pros: alta capacidad de integración y respuesta en tiempo real.
* Contras: más compleja de implementar y requiere infraestructura robusta.