Nombre	Ventajas	Desventajas
P. capas	Una capa inferior puede ser utilizada por diferentes capas superiores. Las capas facilitan la estandarización ya que podemos definir claramente los niveles. Se pueden realizar cambios dentro de la capa sin afectar a otras capas.	No es de aplicación universal. Es posible que sea necesario omitir ciertas capas en determinadas situaciones.
P. cliente – servidor	Es bueno modelar un conjunto de servicios donde los clientes puedan solicitarlos.	Las solicitudes normalmente se manejan en subprocesos separados en el servidor. La comunicación entre procesos genera gastos generales ya que diferentes clientes tienen diferentes representaciones.
P. maestro - esclavo	Precisión: la ejecución de un servicio se delega a diferentes esclavos, con diferentes implementaciones.	Los esclavos están aislados: no hay un Estado compartido. La latencia en la comunicación maestro-esclavo puede ser un problema, por ejemplo, en sistemas en tiempo real. Este patrón sólo se puede aplicar a un problema que se puede descomponer.
P. filtro de tubería	Muestra procesamiento concurrente. Cuando la entrada y la salida consisten en flujos, la eficiencia está limitada por el proceso de filtrado más lento. y los filtros comienzan a calcular cuando reciben datos. Fácil de agregar filtros. El sistema se puede ampliar fácilmente. Los filtros son reutilizables. Puede construir diferentes canalizaciones recombinando un conjunto determinado de filtros.	Gastos generales de transformación de datos al pasar de un filtro a otro. La eficiencia está limitada por el proceso de filtrado más lento.
P. agente	Permite cambios dinámicos, adición, eliminación y reubicación de objetos, y hace que la distribución sea transparente para el desarrollador.	requiere estandarización de las descripciones de servicios. hace que la distribución sea transparente para el desarrollador.
P- igual – igual	Admite computación descentralizada. Altamente robusto ante el fallo de cualquier nodo determinado. Altamente escalable en términos de recursos y potencia informática.	No hay garantía de calidad del servicio, ya que los nodos cooperan voluntariamente. Es difícil garantizar la seguridad. El rendimiento depende del número de nodos.
P. bus de evento	Se pueden agregar fácilmente nuevos editores, suscriptores y conexiones. Efectivo para aplicaciones altamente distribuidas.	La escalabilidad puede ser un problema, ya que todos los mensajes viajan a través del mismo bus de eventos.
P. MVC	Facilita tener múltiples vistas del mismo modelo, que se pueden	Aumenta la complejidad. Puede dar lugar a muchas actualizaciones

	conectar y desconectar en tiempo de ejecución.	innecesarias para las acciones del usuario.
P. blackboard	Fácil de agregar nuevas aplicaciones. Ampliar la estructura del espacio de datos es fácil.	Modificar la estructura del espacio de datos es difícil, ya que todas las aplicaciones se ven afectadas. Puede necesitar sincronización y control de acceso.
P. interprete	Es posible un comportamiento muy dinámico. Bueno para la programabilidad del usuario final. Mejora la flexibilidad, porque reemplazar un programa interpretado es fácil.	Debido a que un lenguaje interpretado es generalmente más lento que uno compilado, el rendimiento puede ser un problema.