**Estilos y Patrones ARQUITECTÓNICOS**

**ESTILOS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estilo Arquitectónico** | **Descripción** | **Atributos Asociados** | **Atributos en conflictos** |
| **Datos Centralizados** | Sistemas en los cuales cierto número de clientes accede y actualiza datos compartidos de un repositorio de manera frecuente | Integrabilidad Escalabilidad Modificabilidad | Desempeño |
| **Flujo de Datos** | El dato ingresa en el sistema, y fluye entre los componentes, de uno en uno, hasta que se le asigne un destino final (salida o repositorio) | Reusabilidad Modificabilidad Mantenibilidad | Desempeño |
| **Llamada y Retorno** | El sistema se constituye de un programa principal que tiene el control del sistema y varios subprogramas que se comunican con éste mediante el uso de llamadas | Modificabilidad Escalabilidad Desempeño | Mantenibilidad  Desempeño |
| **Peer to Peer** | Consiste en un número de procesos u objetos independientes que se comunican a través de mensajes | Modificabilidad Escalabilidad | Desempeño  Integrabilidad |
| **Sistemas basados en Eventos** | Responde a las acciones o acontecimientos generados por un directorio y sus usuarios | Desempeño  Modificabilidad | Mantenibilidad |

**Centrado en Datos:**

* **Repositorio**
* **Pizarra**

**REPOSITORIO**

Todos los datos de un sistema se gestionan en un repositorio central, accesible a todos los componentes del sistema. Los componentes no interactúan directamente, sino a través de un repositorio.

Se usa cuando se tiene un sistema donde los grandes volúmenes de información generados deban almacenarse durante mucho tiempo. También puede usarse en sistemas dirigidos por datos, en los que la inclusión de datos en el repositorio active una acción o herramienta.

**Ejemplo:**

**Base de datos de los ciudadanos de un país**

**REGISTRO CIVIL**

**ONAT**

**ANTECEDENTES**

**PENALES**

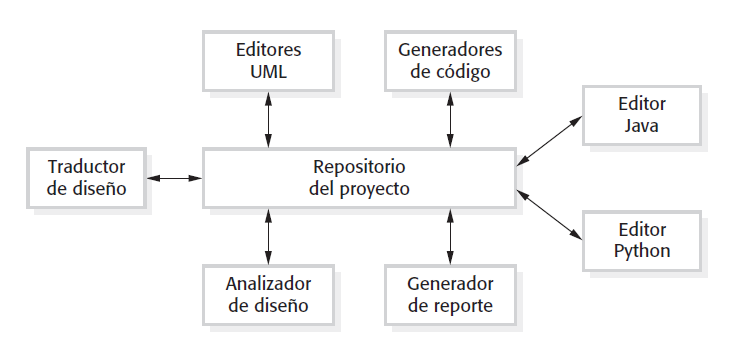
**SISTEMA ELECTORAL**

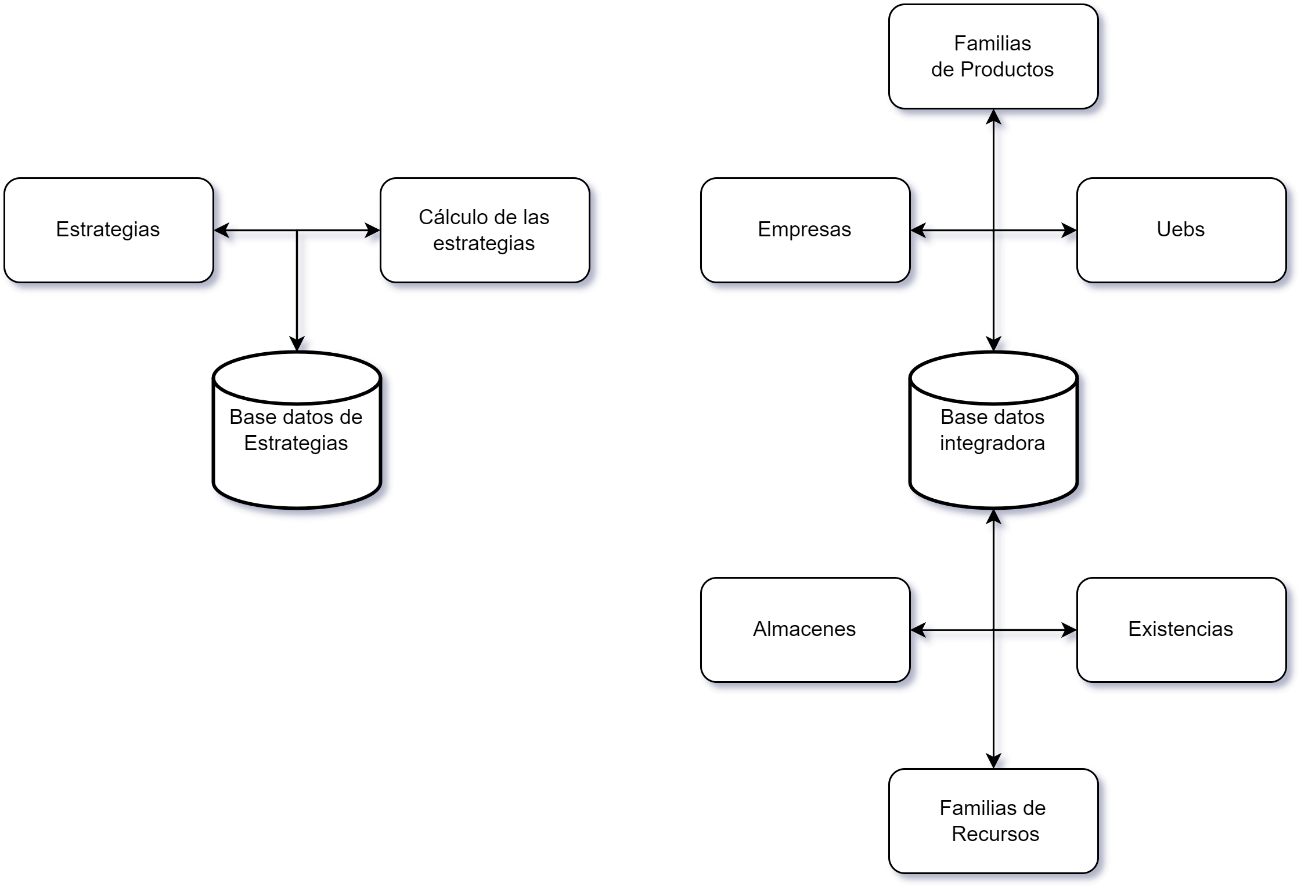
**REGISTRO VEHICULAR**

**INMIGRACIÓN**

**REGISTRO DE HOSPITALES**

**CONSISTE EN UNA BASE DE DATOS A LA QUE ACCEDEN VARIOS SISTEMAS**

****



**PIZARRA O BLACKBOARD**

Sistema multiagentes que contenga múltiples agentes autónomos con capacidad incompleta para resolver el problema global y que al trabajar en forma integrada son capaces de detectar, evaluar y alertar sobre vulnerabilidades de forma temprana, para que se puedan tomar medidas antes de que ocurran hechos delictivos

**Vulnerabilidades percibidas desde los diferentes puntos de acceso**

**AGENTE QUE PERCIBE LAS VULNERABILIDADES**

**AGENTE QUE EVALÚA EL TIPO DE ACCESO NO AUTORIZADO (INTERNO O EXTERNO)**

**AGENTE QUE ALERTA DE MANERA TEMPRANA**

* Posibilita la integración de agentes.
* Adecuado para la resolución de problemas no deterministas.
* Se puede resumir el estado de conocimiento en cada momento del proceso

**Flujo de Datos:**

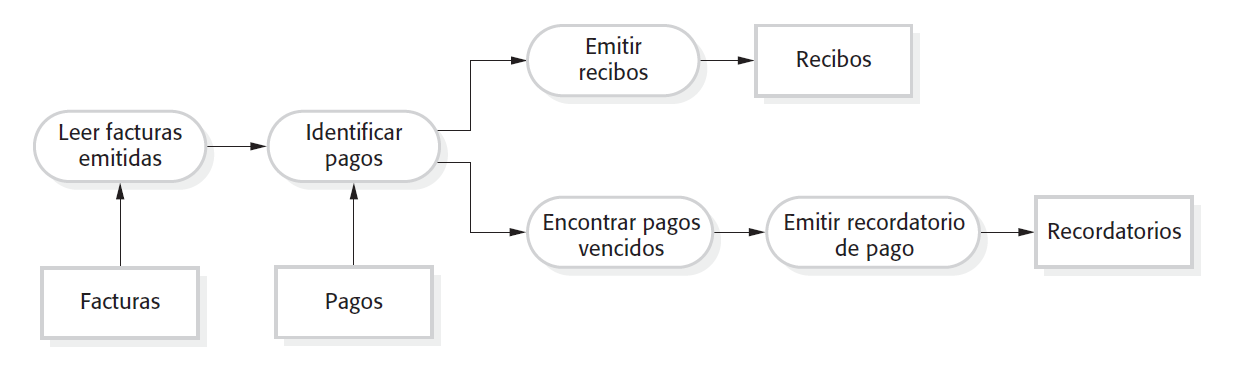
* **Filtros y Tuberías**

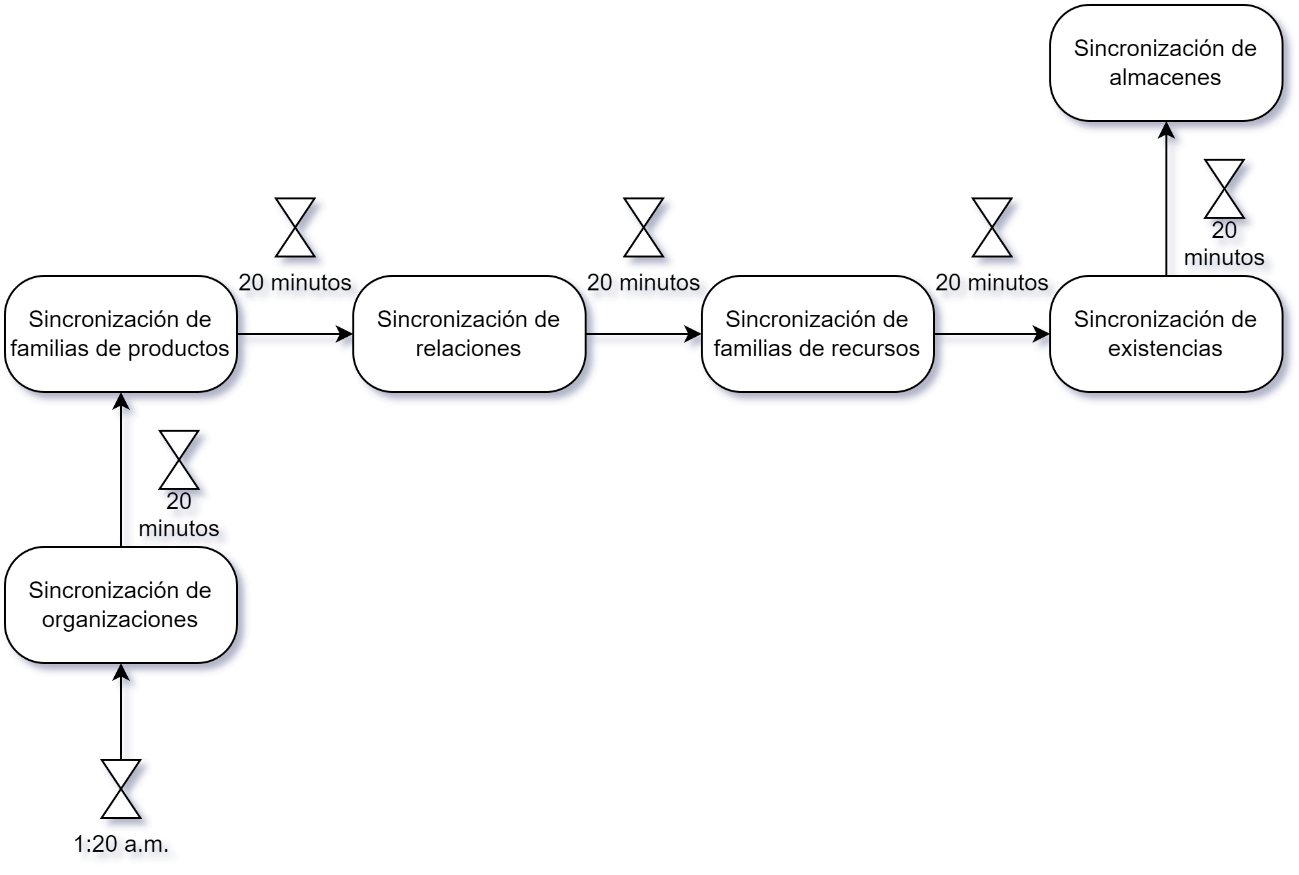
**FILTROS Y TUBERÍAS**

El procesamiento de datos en un sistema se organiza de forma que cada componente de procesamiento (filtro) sea discreto y realice un tipo de transformación de datos. Los datos fluyen (como en una tubería) de un componente a otro para su procesamiento.

Se suele usar en aplicaciones de procesamiento de datos (tanto basada en lotes batch como en transacciones), donde las entradas se procesan en etapas separadas para generar salidas relacionadas. También se usa cuando:

* Se puede especificar la secuencia de un número conocido de pasos.
* No se requiere esperar la respuesta asincrónica de cada paso.
* Se busca que todos los componentes situados corriente abajo sean capaces de inspeccionar y actuar sobre los datos que vienen de arriba (pero no viceversa)





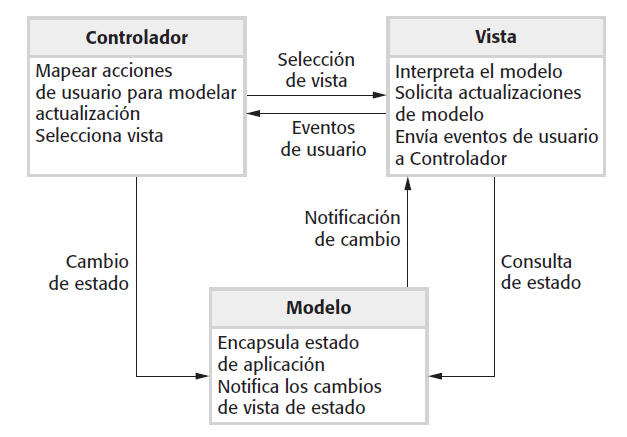
**Llamada y Retorno:**

* **MVC**
* **N-capas**
* **Cliente-Servidor**

**MVC**

Separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente MODELO maneja los datos de sistema y las operaciones asociadas a esos datos. El componente VISTA define y gestiona cómo se presentan los datos al usuario. El componente CONTROLADOR dirige la interacción del usuario y las pasa a Vista y Modelo.

Se usa cundo existen múltiples formas de ver e interactuar con los datos. También se utiliza al desconocerse los requerimientos futuros para la interacción y la presentación.



**N-Capas**

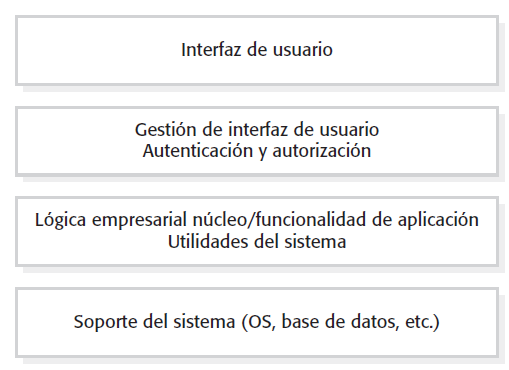
Organiza el sistema en capas con funcionalidad relacionada con cada capa. Una capa da servicios a la capa de encima, de modo que las capas de nivel inferior representan servicios núcleo que es probable se utilicen a lo largo de todo el sistema.

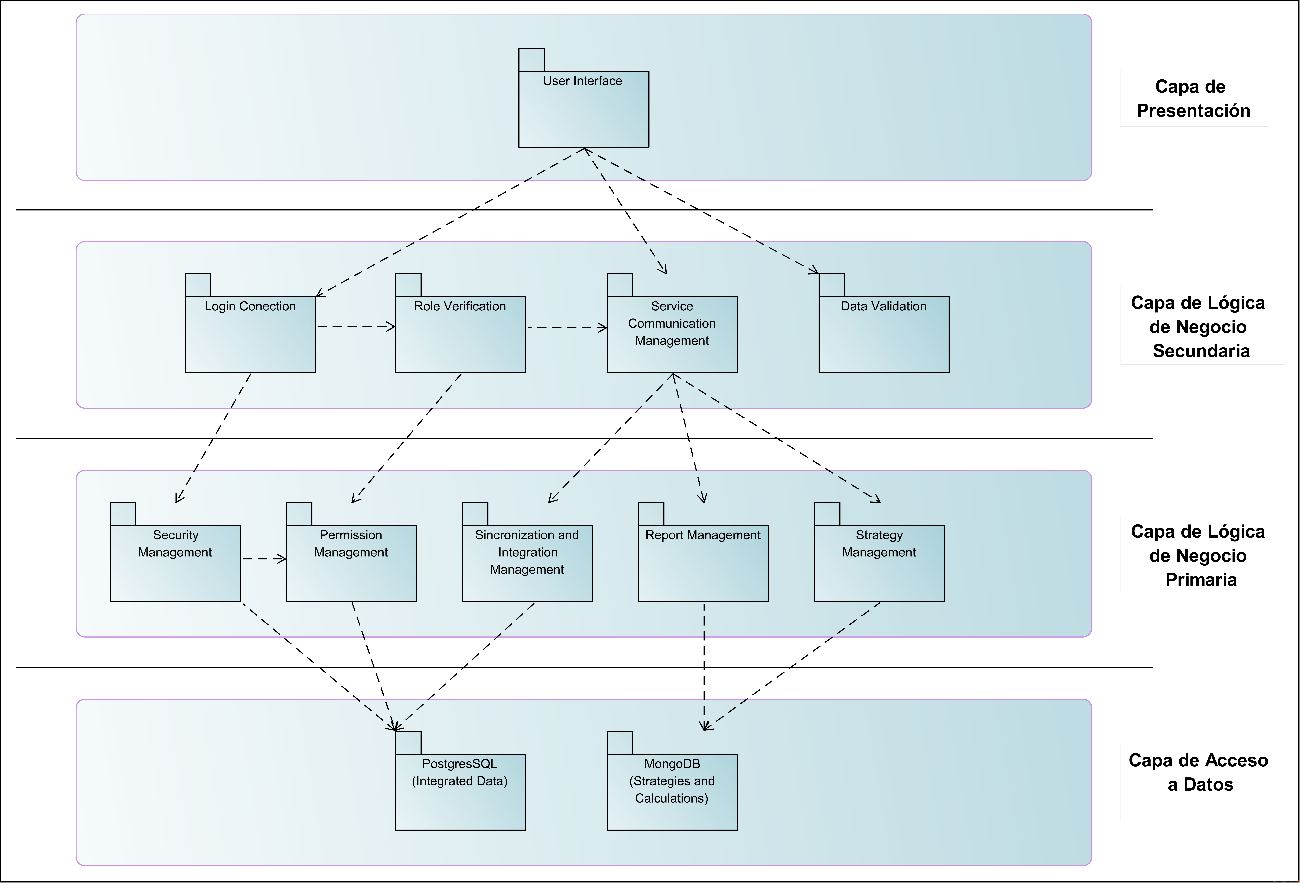
Se usa:

* Al construirse nuevas facilidades encima de los sistemas existentes
* Cuando el desarrollo se dispersa a través de varios equipos de trabajo, y cada uno es responsable de una capa de funcionalidad
* Cuando exista un requerimiento para seguridad multinivel

Fig. : Enfoque Responsabilidad

Fig. :Enfoque Reutilización

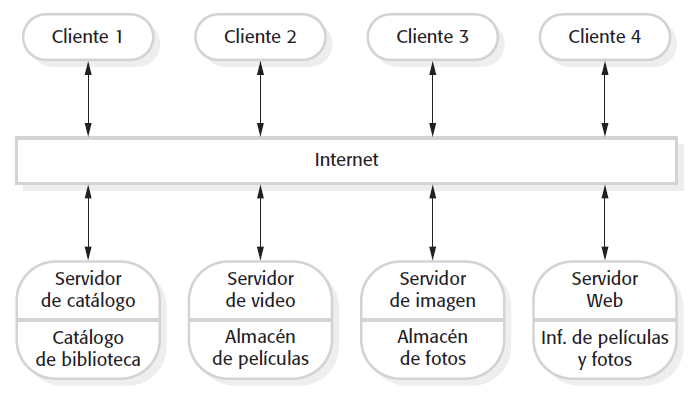




**CLIENTE – SERVIDOR**

La funcionalidad del sistema se organiza en servicios y cada servicio lo entrega un servidor independiente. Los clientes son usuarios de dichos servicios y para utilizarlos ingresan a los servidores.

Se usa cuando, desde varias ubicaciones, se tiene que ingresar a los datos en una base de datos compartida. Como los servidores se pueden replicar, también se usan cuando la carga de un sistema es variable.



**Peer to Peer:**

* **SOA**

**SOA**

Todas las funciones están definidas como servicios independientes *con* interfaces invocables que pueden ser llamados en secuencias bien definidas para formar los procesos de negocio. Separa los procesos de negocio de las funciones automatizadas y organiza estas últimas en módulos individuales catalogados en un diccionario de servicios que permiten su utilización por parte de toda la organización.

Se usa:

* Construcción de aplicaciones con múltiples servicios y una interfaz única.
* Construcción de aplicaciones en la nube.
* Comunicación basada en mensajes.
* Funcionalidad con independencia de plataformas.
* Necesidad de establecer servicios federados.
* Exposición de servicios con independencia del conocimiento por parte del cliente de su interfaz.
* Adecuado para escenarios de interoperabilidad e integración.

