

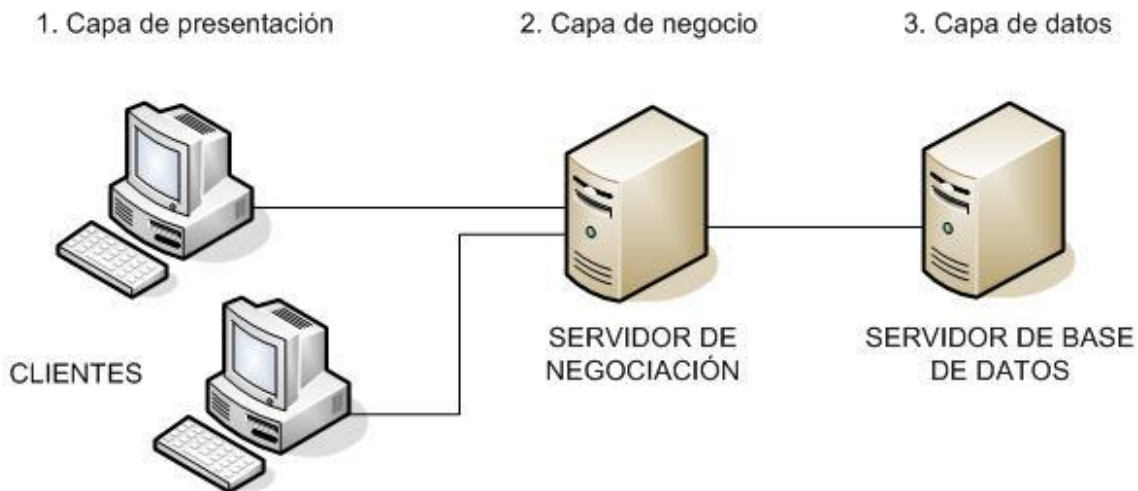
Arquitectura De Desarrollo De Software a Utilizar (Proyecto)

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo, o, por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente.

En [GS94] Garlan y Shaw definen el estilo en capas como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. (más de 100 patrones o variantes de este estilo).

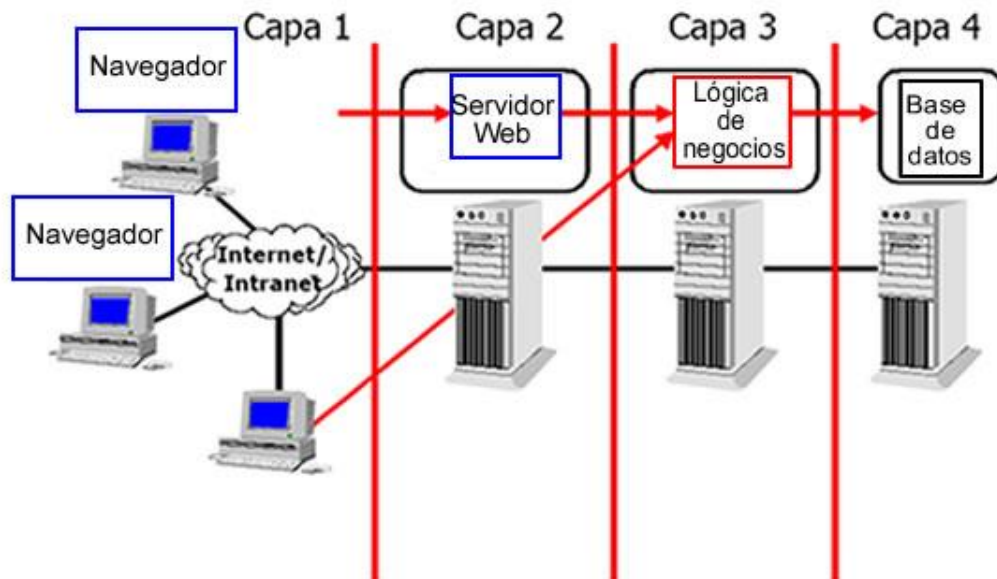
EC – Arquitectura Cliente / Servidor

Un componente servidor, que ofrece ciertos servicios, escucha que algún otro componente requiera uno; un componente cliente solicita ese servicio al servidor a través de un conector. El servidor ejecuta el requerimiento (o lo rechaza) y devuelve una respuesta.



Ventajas de los Modelos en Capas

- El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.
- El estilo admite muy naturalmente optimizaciones y refinamientos.
- Proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas.



RESUMEN MODELO EN CAPAS

La arquitectura basada en capas como bien sabemos se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

Estas son las características porque voy a usar esta arquitectura:

- Porque describe la descomposición de servicio de forma que la mayoría de la interacción ocurre solamente entre capas vecinas
- Porque las capas que se utilizaran en una aplicación pueden residir en la misma maquina física (misma capa) o puede estar distribuido sobre diferentes computadoras (n-capas).
- Porque los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.
- Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.