El patrón arquitectónico por capas ayuda a estructurar las aplicaciones que pueden ser descompuestas en grupos de tareas en la cual cada grupo de tareas es un nivel particular de abstracción.

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.

Este patrón cuenta con diferentes capan que envuelven funcionalidades específicas del sistema y tienen unas interfaces de comunicación definidas mediante protocolos de comunicación. Originalmente este patrón supone que las comunicaciones se realizan únicamente entre capas adyacentes y que las comunicaciones entre elementos de una capa solo pueden ser entre elementos de la misma. Si esto no se cumple, el estilo deja de ser puro y se pierden algunas de sus características

Los sistemas de información usados en los sistemas de dominio de negocio, usualmente utilizan arquitecturas de dos capas. La capa inferior es la base de datos que contiene los datos de la compañía. En esta arquitectura muchas aplicaciones trabajan de manera concurrente encima de esta capa de datos, para cumplir diferentes tareas. Por el acoplamiento tan alto que esto genera entre la interface de usuario y los datos, se introduce una tercera capa en medio de estas, llamada la capa de dominio, que modela la estructura conceptual del domino del problema. De igual manera es necesario subdividir esta capa, para final mente tener una arquitectura de cuatro capas:

* Presentación
* Lógica de la aplicación
* Dominio
* Datos

Beneficios:

Reutilización de capas: Se puede dar en la medida en que cada capa tenga un nivel de abstracción bien definido y unas interfaces documentadas y bien definidas igualmente. Si esta caja negra está desarrollada correctamente, puede reducir dramáticamente el esfuerzo en desarrollo y la cantidad de defectos

Soporte de estandarización: Unos niveles de abstracción claramente definidos y comúnmente aceptados, permiten el desarrollo de tareas e interfaces estandarizadas

Posibilidad de intercambio: Se pueden intercambiar diferentes implementaciones de la misma capa que sean semánticamente equivalentes, sin realizar gran esfuerzo.