Arquitectura en Capas

Estilo arquitectónico

La arquitectura en capas es una de las más utilizadas, no solo por su simplicidad, sino porque también es utilizada por defecto cuando no estamos seguros que arquitectura debemos de utilizar para nuestra aplicación.

La arquitectura en capas consta en dividir la aplicación en partes, con la intención de que cada una tenga un rol muy definido, como podría ser, una capa de presentación (UI - User Interface o Interfaz de Usuario), una capa de reglas de negocio (servicios) y una capa de acceso a datos (DAO - Data Access Objects o Objetos de Acceso a Datos), sin embargo, este estilo arquitectónico no define cuántas capas debe de tener la aplicación, sino más bien, se centra en la separación de la aplicación en partes distintas de acuerdo a su funcionalidad principal (aplica el principio SoC - Separation of Concerns o Separación de Responsabilidades).

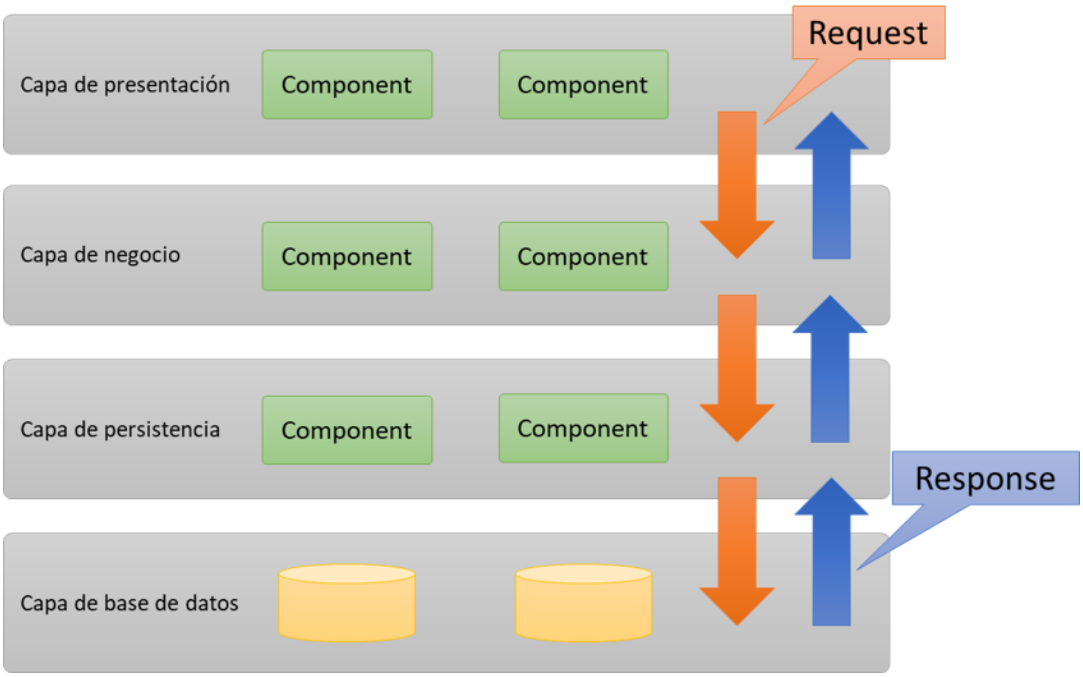
En la práctica, la mayoría de las veces este estilo arquitectónico es implementado en 4 capas, **presentación**, **negocio**, **persistencia** y **base de datos**.

Arquitectura en capas

En ocasiones podemos encontrarnos con sistemas en los cuales la capa de negocio y persistencia se combinan en una sola, es decir que la lógica de persistencia está incrustada dentro de la capa de negocio. Esto **no se considera una buena práctica**, ya que eleva el nivel de acoplamiento de nuestro sistema.

Como se estructura una arquitectura en capas

En una arquitectura en capas, todas las capas se colocan de forma horizontal, de tal forma que cada capa solo puede comunicarse con la capa que está inmediatamente por debajo, por lo que, si una capa quiere comunicarse con otras que están mucho más abajo, tendrán que hacerlo mediante la capa que está inmediatamente por debajo. Por ejemplo, si la capa de presentación requiere consultar la base de datos, tendrá que solicitar la información a la capa de negocio, la cual, a su vez, la solicitará a la capa de persistencia, la que a su vez, la consultará a la base de datos, finalmente, la respuesta retornará en el sentido inverso hasta llegar la capa de presentación.



Comunicación entre capas.

La separación de la aplicación en capas busca cumplir con el principio de SoC, de tal forma que cada capa se encargue una tarea muy definida.

Por ejemplo, la capa de presentación solo se preocupa por presentar la información de forma agradable al usuario, pero no le interesa de dónde viene la información ni la lógica de negocio que hay detrás. En su lugar, solo sabe que existe una capa de negocio que le proporcionará la información.

Por otra parte, la capa de negocio solo se encarga de aplicar todas las reglas de negocio y validaciones, pero no le interesa cómo recuperar los datos, guardarlos o borrarlos, ya que para eso tiene una capa de persistencia que se encarga de eso.

Por otro lado, la capa de persistencia es la encargada de comunicarse a la base de datos y ejecutar las instrucciones necesarias para consultar, insertar, actualizar o borrar registros y

retornarlos en un formato independiente a la base de datos. De esta forma, cada capa se preocupa por una cosa y no le interesa cómo hace la capa de abajo para servirle los datos que requiere.

Respetar el orden de las capas es muy importante, y saltarse una capa para ir directamente a una más abajo suele ser un grave error, ya que si bien es posible hacerlo, empezamos a crear un desorden sobre el flujo de comunicación, lo que nos puede llevar al típico anti-patrón conocido como código espagueti, el cual consiste en que empezamos a realizar llamadas desde cualquier capa a otra capa, haciendo que el mantenimiento se vuelva un verdadero infierno.