**NOMBRES:** Fabián Romero

Iván Ballesteros

Oscar Suarez

Diego Pedraza

Germán Piñeros

Andrés Tuta

**TALLER ARQUITECTURA EN CAPAS**

1. **Defina capas en este estilo de arquitectura.**

Una capa es un grupo de módulos que ofrecen un conjunto de servicios a otras capas. Las capas son relacionadas. Las capas están relacionadas entre sí por la relación estrictamente ordenada permitida a utilizar.

1. **Mencione los beneficios de la arquitectura en capas.**

-Promueve modificación y da portabilidad al sistema de software

-Maneja la complejidad y facilita la comunicación de la estructura del código a los desarrolladores.

-Promueve la reutilización de código

-Logra la separación de “concerns”

1. **Explique la restricción en la relación entre capas “permitido usar”**

Se permite usar, la cual es una especialización de las relaciones genéricas de dependencia. El diseño debería definir las reglas de uso por capas. Por ejemplo una capa puede usar cualquier capa inferior. Cualquier excepción es permitida.

1. **Algunos arquitectos / arquitectas de software diseñan sistemas que en realidad no constituyen arquitectura en capas. Explique esta situación con un ejemplo.**

Un error común de la arquitectura por capas es que una capa inferior usa o implementa una superior.

Otro error es que una capa se comunica con capas superiores o con capas que se encuentran varios niveles abajo. Ya que lo permitido es que una capa se comunique con la que está inmediatamente abajo.

A

B

C

1. **Los módulos presentes en una capa se comunican a través de interfaces con los módulos en la capa siguiente. Esto implica la aplicación del principio arquitectónico de “ocultamiento de información (information hiding) Explique su funcionamiento y ventajas.**

Una capa es una aplicación del principio de ocultamiento de información. La teoría es que un cambio a una capa inferior puede ser ocultado detrás de su interfaz y no afectará a las capas superiores.

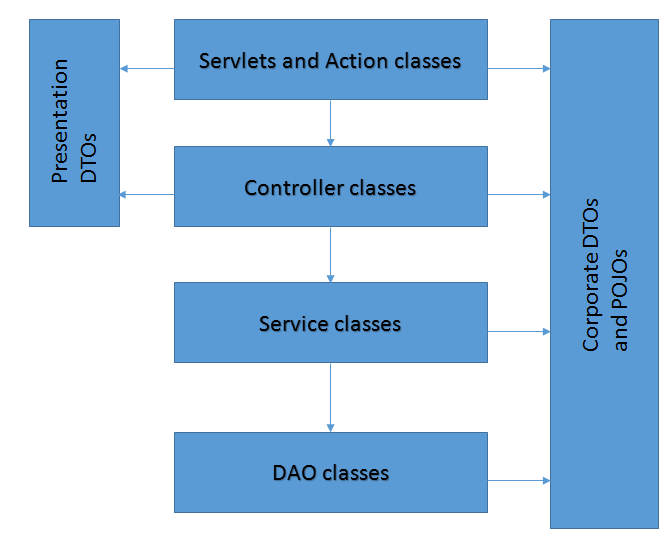
1. **Explique la implementación de la portabilidad en esta arquitectura.**

La interfaz no debería exponer funciones que dependen de una plataforma en particular, estas funciones deben ser ocultadas detrás de otra interfaz más abstracta que es independiente de la plataforma.

1. **Explique la diferencia entre capas (layers) y niveles (tiers).**

El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico, mientras que el término nivel corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo un diseño de tres capas MVC puede estar soportado por un solo servidor, por lo cual dicha arquitectura solo posee un nivel. Sin embargo esta arquitectura, aunque bien diseñada lógicamente puede no estar acorde con las exigencias a la aplicación, pues un solo ordenador podría estar sobrecargado, por lo cual es importante desplegar el software en varias capas.

1. **Clements describe dos ejemplos de arquitectura por capas: el sistema operativo UNIX y una aplicación WEB basada en JEE. Explique cualquiera de las dos.**

****

Modelo de capas de una aplicación WEB basada en JEE.

Las capas superiores contienen los servlets y JSF action clases responsables de la interfaz de usuario. Las clases controlador manejan las operaciones de usuario interactuando con las clases de servicio del negocio las cuales capturan el núcleo de la lógica del negocio. Las clases de la capa más baja son las encargadas del acceso a la base de datos haciendo uso de objetos DAO (Data Access Object).

Las capas laterales contienen DTOs y POJOs que son usados por las otras capas para guardar y transferir datos.