 **INSTITUTO TECNOLOGICO**

**DE CULIACAN**

**ACTIVIDAD #5**

**Presentación del articulo Introducción a la   
Arquitectura de Software de Garlan & Shaw**

**FILTROS Y MANGUERAS:**

Como su nombre lo dice este estilo de arquitectura posee dos elementes esenciales, los llamados filtros, que vendrían siendo los componentes, y las mangueras, estas dirigen el flujo de los datos tanto de entrada como de salida en cada uno de los filtros. Los componentes tiene la capacidad de leer datos, procesarlos y entregar un resultado completo e incremental de las peticiones hechas por los datos de entrada, las salidas (al igual que las entradas) pasan por las llamadas mangueras (como se mencionó anteriormente dirigen el flujo de los datos) convirtiéndose en una entrada de datos al siguiente componente (filtro). El flujo de datos que viaja por las mangueras solo puede darse por un sentido, lo que hace de este estilo no apto para aplicaciones que interactúan con el usuario. Este estilo posee diferentes tipos de mangueras, cada una con una función diferente, por ejemplos las “acotadas”, que sirven para restringir la cantidad de datos que pasaran por la manguera.

Cabe mencionar que este estilo de arquitectura de software posee varias variantes importantes que debemos conocer, como por ejemplo es importante tener en cuenta que los filtros no deben compartir un estado con otro filtro, pues estos son independientes. Por otra parte los filtros no pueden identificar los componentes que se encuentran a los extremos de las mangueras, pues no saben identificar la identidad del componente predecesor ni del sucesor.

**ORIENTADO A OBJETOS:**

En este estilo de arquitectura podemos ver a los componentes como objetos, los cuales pueden interactuar entre sí por medio de las funciones e invocaciones que estos mismos realizan. Con dicho estilo la manipulación de datos se da mediante un conjunto de rutinas que permiten acceder a los datos que se desean manejar. Cabe mencionar que cada uno de los objetos en el sistema tiene la responsabilidad de preservar lo que este representa y ocultarla a otros objetos del mismo sistema, esto sirve para poder cambiar la aplicación sin que se vea afectado un tercero (o bien un cliente). Aquí encontramos una característica muy particular, pues como estamos manejando objetos que encapsulan un tipo de datos abstracto, se puede permitir que estos realicen tareas concurrentes y a su vez contar con más de una interfaz.

**BASADA EN EVENTOS, INVOCACION IMPLICITA:**

Este estilo hace referencia a la variación en la invocación implícita, la integración reactiva y la emisión selectiva. Aquí se destaca la invocación implícita pues un componente puede difundir uno o más eventos, lo cual proporciona una colección de procedimientos y un conjunto de eventos. También cabe mencionar que un componente puede registrar procedimientos con eventos del sistema, lo cual provoca que dichos eventos sean anunciados en tiempo de ejecución. Sin embargo cuando un evento se lleve acabo los locutores de dicho evento no sabrán cuales de los componentes se verán afectados por las acciones producidas por dicho evento, por lo que los componentes que posiblemente se ven afectados no podrán prevenir el posible orden de procesamiento y lo que producirá el evento. Por otra parte una ventaja de este estilo que se puede reutilizarse, pues cualquier componente se puede introducir al sistema registrándolo para los acontecimientos de dicho sistema.

**SISTEMA DE CAPAS:**

Dentro de este estilo se especifica que cada capa debe estar organizada jerárquicamente, cada elemento (en este caso las capas) proporciona un servicio a la capa, mientras que le sirve como cliente a la capa inferior, este sistema al contrario del estilo de filtros y mangueras, es apto para la interacción con el usuario. Pues por el mismo canal puede enviar y recibir datos. La interacción entre las capas se define por un protocolo de conexión, dicho protocolo restringe ciertas interacciones con las capas adyacentes del sistema. El protocolo de comunicación más conocido es el de “comunicación por capas”; cada capa proporciona un sustrato para la comunicación en algún nivel de abstracción, por ejemplo el nivel más bajo de interacción es la conexión al hardware, entre otras. El estilo en capas al igual que los otros estilos, posee varias características y propiedades, por ejemplo; aquí se permite particionar un problema complejo en una secuencia de pasos incrementales, lo cual ayuda hacer de este problema más sencillo de resolver. Por otro lado los cambios que se realicen cualquiera de las capas afecta a la mayoría de las otras capas existentes en el sistema.

**PIZARRA:**

Debemos mencionar que en el estilo pizarra existen dos tipos de componentes:

-Centro estructurado de datos.

-Colección independiente de componentes.

La primera se destaca por que representa el estado actual de dicha estructura, por otra parte el segundo se caracteriza por que los componentes operan en un almacén central de datos. La selección de alguna de esas dos disciplinas varía, por ejemplo si el estado actual de la estructura de datos central es el principal desencadenante de la selección de los procesos para ejecutar el repositorio puede ser una pizarra. Por otra parte si los tipos de transacciones en un flujo de entrada destacan la selección de procesos para ejecutar el repositorio puede ser una base de datos tradicional. En la primera disciplina (pizarra) se destacan tres aspectos principales: fuentes de conocimiento, estructura de la pizarra y el control.

En las fuentes de conocimiento encontramos la aplicación se separa en partes diferentes de conocimiento. En lo que es la estructura de la pizarra podemos destacar que se organizan de forma jerárquica los estados del problema, para así resolverlo. Por último el control el guiado por el paso anterior, depende del estado en que se encuentre la pizarra.

**Referencias:**

Act 5 Introduction+to+SA+Garlan