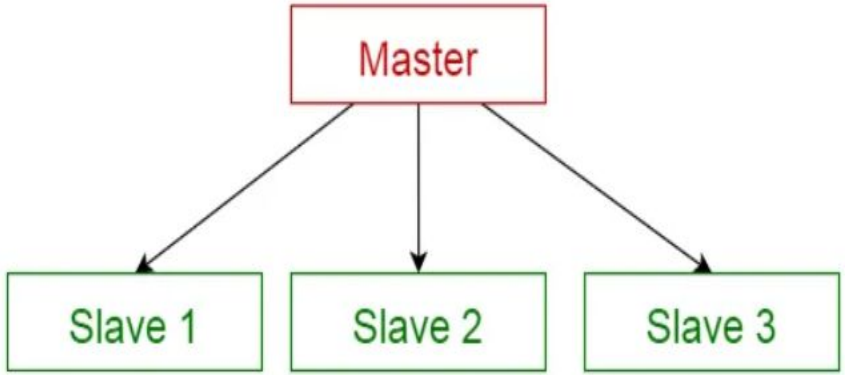
Maestro-Esclavo

Sabrina Brenes Hérnandez. C01309
Gabriel Chacón Garro. C02063
Lizeth Corrales Cortés. C02428
Andrés Matarrita Miranda. C04668
Esteban Mora García. C05126

Definición

Se tiene un componente maestro que distribuye el trabajo entre los componentes esclavos, se cuenta con distintas formas de distribución de los procesos para realizar las tareas de manera optimizada.



Ventajas

- Debido a que hay un componente maestro que distribuye la información entre los nodos esclavos, esta arquitectura tiene gran capacidad de escalabilidad, además de ser un proceso relativamente sencillo.
- Cuando se requiere hacer aplicaciones multitareas, los nodos esclavos facilitan este proceso y mejoran de gran manera el rendimiento general.
- Si un nodo esclavo falla es fácilmente reemplazable por uno nuevo.
- Un nodo esclavo puede servir de respaldo para otro en caso de pérdida de datos.

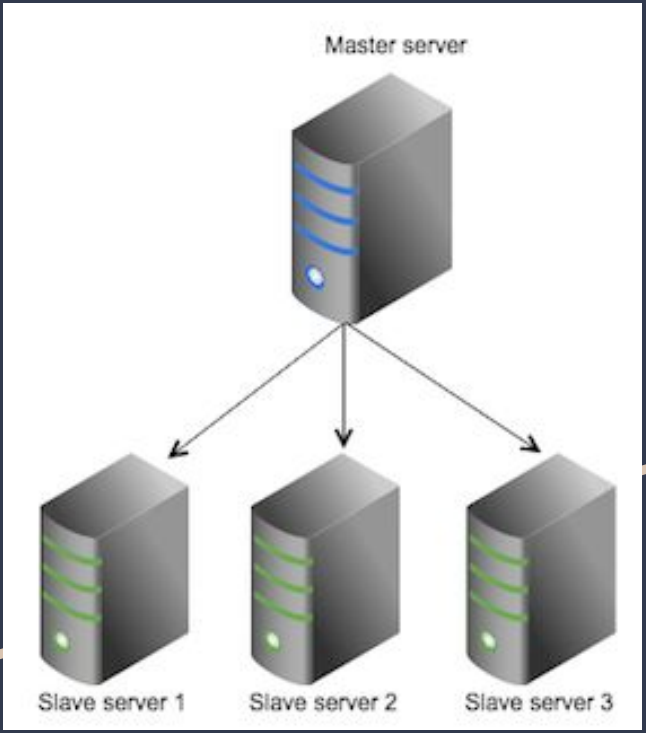
Desventajas

- **Problemas en confiabilidad:** Posee un único punto de fallo, dado que si el maestro falla el resto de componentes queda inoperante.
- **Concurrencia**: Se debe poseer un buen manejo de concurrencia, pues podrían presentarse condiciones de carrera al varios hilos manipular recursos compartidos.
- Se pueden dar sobrecargas en el sistema por parte del procesador maestro.
- **Problemas de sincronización:** es posible que el bucle esclavo inicie su ejecución primero que el bucle maestro. Por lo tanto, se debe inicializar el bucle maestro antes que los bucles esclavos inicien su ejecución.

¿Cuándo usar?

- Cuando se tiene 2 o más procesos que necesiten ejecutarse de manera simultánea y continuamente, pero estos realizan sus tareas a velocidades distintas.
- Cuando se desarrollan aplicaciones multitarea, pues su funcionalidad multi-bucle, provoca mejoras en la gestión de tiempo.
- Cuando la paralelización de tareas es requerida.

Ejemplo de sistema



Se tienen casos en los que en los que se puede emplear este patrón, por ejemplo cuando se tiene una impresora que controla la cola de impresión de las demás impresoras, esta se encarga de distribuir y controlar el trabajo, se puede aplicar también con cámaras también tienen a carga otra cantidad de cámaras y por último en el procesamiento de imágenes, el maestro le dá a los esclavos que parte de la imágen procesar y luego este junta toda la imágen.

Pipe-Filter

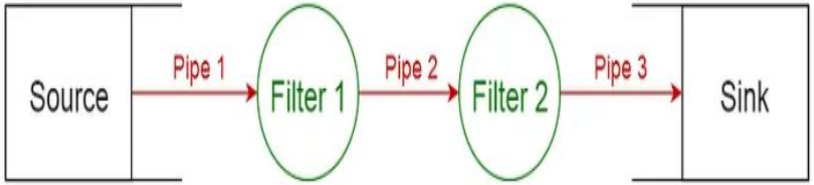
Tuberías y Filtros

Sabrina Brenes Hérnandez. C01309
Gabriel Chacón Garro. C02063
Lizeth Corrales Cortés. C02428
Andrés Matarrita Miranda. C04668
Esteban Mora García. C05126

Definición

Tiene entidades independientes:

- Filtro(Filters): son componentes que hacen transformaciones en los datos y procesan el input que reciben.
- Tuberías(Pipes): sirven como conectores para el flujo de datos de la aplicación, cada tubería se conecta con el siguiente componente que transforma los datos.
- Cada filtro puede consumir datos o producir datos, pueden actuar de forma concurrente y la salida de un filtro es la entrada del siguiente.



Ventajas

- Asegura conexiones flexibles entre los filtros.
- Esa característica flexible permite el cambio de filtros sin hacerle modificaciones a otros.
- Permite el procesamiento en paralelo.
- Los usuarios no tienen que saber la lógica en los filtros.
- Cada filtro es reusable y puede usarse una y otra vez.
- Es útil para procesos que requieren pasos muy claros.
- Tiene una separación de asuntos natural, ya que cada filtro se implementa de forma independiente.

Desventajas

- No es recomendable cuando hay mucha interacción entre los usuarios y el sistema.
- Puede no ser apropiado cuando se trata de procesos de mucha duración.
- Es ineficiente porque pasa todo el flujo de datos por todos los filtros y unos pueden no necesitarlos.
- Puede tener cuellos de botella o deadlocks al pasar tanta información por los filtros.

¿Cuándo usar?

El uso del pipe filter es especialmente útil cuando se trabaja con datos dinámicos y se necesita una presentación adecuada en la vista. En lugar de manipular los datos en el componente, se puede utilizar el pipe filter para transformar los datos justo antes de mostrarlos a la vista. Esto hace que el código sea más fácil de mantener y reduce la cantidad de lógica necesaria en el componente.

Ejemplo de sistema

Un ejemplo de sistema que utiliza el filtro de tubería podría ser un sistema de búsqueda de productos en un sitio web de comercio electrónico. El usuario puede ingresar una palabra clave en un campo de búsqueda y el sistema devolverá una lista de productos que coincidan con esa palabra clave.

En este sistema, el filtro de tubería podría usarse para filtrar y ordenar los resultados de la búsqueda antes de mostrarlos al usuario.