

ASIGNATURA: Programación Distribuida

AÑO:

NOMBRE: Juan Sebastián Quishpe

CALIFICACIÓN:

FECHA: 28/02/2023

TEMA: Fault Tolerance

## Fault Tolerance

La tecnología Fault Tolerance se refiere a la capacidad de un sistema informático electrónico o de red para mantener su funcionamiento sin interrupciones, incluso en presencia de fallas en uno o más de sus componentes. Este tipo de tecnología también abarca la posibilidad de interrupciones en el servicio (lógico) debido a errores lógicos o de software. El objetivo principal de la tolerancia a fallas es evitar que se produzcan fallas catastróficas como resultado de la existencia de un único punto de falla en el sistema.

El término Fault Tolerance es un concepto familiar para los profesionales de tecnología de la información, el cual se ha utilizado desde los años 50 para describir sistemas que deben seguir funcionando, sin importar las circunstancias. En sus primeras etapas, los planes de tolerancia a fallas incluían alertas que notificaban al personal cuando algo estaba a punto de fallar, y se requería su intervención inmediata. Sin embargo, los planes modernos involucran copias de seguridad y redundancias que permiten al equipo seguir trabajando mientras el sistema permanece en línea.

Un sistema que integre tolerancia a fallas debe:

- **Eliminar:** No permite un solo punto de falla, el sistema funciona sin parar aún si se le debe realizar reparaciones.
- **Aislar:** Se debe eliminar el elemento defectuoso del sistema en lugar de dejar que provoque una cascada de problemas.
- **Comprometer:** Cuando se completa la reparación, el elemento dañado vuelve a estar en línea sin interrupciones notables.

Existen múltiples técnicas de tolerancia a fallas que incluyen:

- **Replicación:** Es una técnica compleja que involucra múltiples versiones idénticas de sistemas y subsistemas, garantizando que siempre proporcionen resultados idénticos. En caso de que se identifiquen resultados diferentes, se utilizará un procedimiento para identificar el sistema defectuoso o para verificar si un sistema muestra un resultado diferente, lo que indica que está defectuoso. Esto puede tener lugar a nivel de componente o a nivel de sistema informático completo.
- **Diversidad:** Puede proporcionar tolerancia a fallas a través de elementos alternativos, como respaldo. Pero, esto puede resultar en una copia de seguridad con una capacidad inferior a la fuente principal.
- **Redundancia:** Esto puede lograrse mediante el uso de elementos que asumen la función y el rendimiento del sistema en caso de que el elemento principal falte. Alternativamente se puede disponer de un sistema informático completo como copia de seguridad en caso de falla.

### Poult Tolerance en Web Apps

La tolerancia a fallas garantiza que las aplicaciones en línea estén disponibles en todo momento. La distribución de carga de trabajo es crucial para las aplicaciones web y se logra mediante la implementación de múltiples servidores que pueden compartir la carga de manejo eficiente. Esto no solo mejora la capacidad de respuesta del sistema, sino que también proporciona una estrategia efectiva para superar la falla de un servidor. En resumen, la tolerancia a fallas y la distribución de carga trabajan en conjunto para garantizar la disponibilidad constante de las aplicaciones en línea y mejorar la experiencia del cliente.

## Fault Tolerance en Microprofile

Es importante implementar la tolerancia a fallas en Microprofile porque los sistemas modernos son cada vez más complejos y están compuestos por numerosos servicios interconectados. La falla de uno de estos servicios puede tener un impacto significativo en la funcionalidad general del sistema, lo que puede afectar la disponibilidad y la confiabilidad del servicio.

La tolerancia a fallas puede mitigar estos problemas al permitir que los servicios afectados se recuperen automáticamente o se conecten a alternativas en caso de fallas. Esto garantiza que el sistema en su conjunto sigue funcionando sin interrupción y que los usuarios finales tengan una experiencia fluida y sin interrupciones.

Eclipse Microprofile Fault Tolerance proporciona una solución simple y flexible para crear microservicios tolerantes a fallas. Ofrece las siguientes políticas:

- Timeout
- Retry
- Fallback
- Bulkhead
- Circuit Breaker
- Asynchronous

Estas son etiquetas o anotaciones que se deberían agregar a los métodos o clases.

## Asynchronous

Significa que la ejecución de la solicitud del cliente entrará en un hilo separado, lo que indica que la solicitud no bloqueará el hilo principal de la aplicación. Esto es especialmente útil cuando se manejan solicitudes de larga duración o cuando se espera un alto volumen de solicitudes.

## Timeout

Timeout se utiliza para establecer un límite de tiempo máximo para la ejecución de un método o función, puede resultar útil para evitar que una solicitud se bloquee indefinidamente si la respuesta del servidor tarda demasiado en llegar, es decir es bastante útil en situaciones en las que los recursos son limitados y es importante manejar el tiempo de espera de las solicitudes.

Si el método o función no se completa en el tiempo indicado en la anotación, se lanza una excepción Time Out Exception.

## Retry

Esta anotación permite especificar la cantidad de veces que se debe intentar una operación en caso de que falle debido a una excepción. La anotación se puede aplicar a un método o a una clase para controlar los intentos de ejecución de un método cuando se produce una falla. Retry acepta varios parámetros que permiten configurar el funcionamiento de reinicios, como el número máximo de intentos permitidos, la duración del intervalo de espera entre intentos y el tipo de excepción que se debe considerar para reiniciar la operación.

## Fallback

La anotación Fallback se utiliza para proporcionar una alternativa a una operación que ha fallado. En caso de que la operación principal falte, se invoca la operación fallback para proporcionar una respuesta alternativa o mitigar el impacto del fallo. La anotación se puede aplicar a métodos CDI y se debe proporcionar el método de fallback en la misma clase que el método original y debe tener la misma firma.

ASIGNATURA: Programación Distribuida

AÑO:

NOMBRE: Juan Sebastián Quishpe

CALIFICACIÓN:

FECHA: 28/02/2023

TEMA: Fault Tolerance

## Bulkhead

Sé utiliza para proteger una sección de código de una sobrecarga excesiva de solicitudes y para evitar que un error en una parte del sistema afecte a otras partes. Permite limitar el número de hilos de ejecución que se asignan a un recurso en particular y separar las solicitudes en grupos de recursos independientes para que los errores o sobrecarga no se propaguen al resto del sistema.

Bulkhead se puede aplicar a un método o a una clase y se puede configurar para especificar el número máximo de hilos de ejecución, y de más se puede configurar para especificar un modo de espera si se alcanza el límite máximo de hilos de ejecución.

## Circuit Breaker

Esta anotación se aplica a un método y establece un umbral para el número de errores que puede ocurrir antes de que se abra el circuito. Cuando se alcanza este umbral, el circuito se abre y se evita que se realicen llamadas adicionales al componente que ha fallado durante un periodo de tiempo determinado. Durante este periodo, el circuito permanece abierto y se utiliza una estrategia de Fallback para proporcionar una respuesta alternativa al cliente. Después de que expire el tiempo de espera, el circuito se cierra nuevamente y las llamadas al componente se reenvían.