Prueba Consigna para Almundo.com

Presentado por: Andrés Felipe García Cifuentes

Documento anexo.

En este breve documento de referencia encontrará una breve explicación con la solución planteada para la prueba.

Solución:

- Para dar solución al problema de concurrencia planteado en el enunciado se opta por hacer uso de la interface **Callable** y **Futuros** en java**,** la clase **Dispatcher** implementa la interface **Callable** haciendo de esta clase potencial para llamados concurrentes.
- Para hacer manejo de los 10 llamados concurrentes se usó el api de **ExecutorService**, este permite crear un número determinado de hilos de ejecución mediante el método **newFixedThreadPool.**
- El método **dispatchCall** está encargado de recibir y asignar la llamada a un empleado dependiendo la prioridad de atención, abriendo para esto un hilo de ejecución y retornando un Futuro con la llamada a ser atendida.
- Para manejar los distintos tipos de empleados se crea una clase abstracta llamada **Employee** y de esta heredan las clases **Operator, Supervisor y Director**, adicionalmente se crea la clase Consigna que maneja la lista de empleados disponibles para atender llamadas.
- Para manejar las llamadas se crea la clase **Call** cuyo método principal answer asigna aleatoriamente una duración entre 5 y 10 segundos y se da por terminada.
- La prueba unitaria requerida en la prueba se encuentra en la clase **DispatcherTest** en el Test **requiredTestTenCalls.**

Puntos Extra/Plus

- Para dar solución a la falta de empleados para responder llamadas se opto por una Cola de llamadas creada en la clase Consigna, las llamadas se encolan en el momento que no se encuentra un empleado disponible para contestar, siguiente a eso se da un tiempo de espera de 7 segundos para retomar la primera llamada insertada en la cola y verificar si existe un empleado libre para tomarla de lo contrario se repite el proceso de encolamiento y de espera.
- Para manejar más de 10 llamadas concurrentes se delega la responsabilidad al **ExecutorService** que va realizando las peticiones a medida que tiene hilos disponibles.
- Se realizan más test unitarios para la clase **Dispatcher** y la clase **Call.**

Muchas gracias.