Almundo

Prueba Desarrollador java BackEnd

Edgar Fabián Ibáñez Jiménez

**Solución general**: Se buscó que las llamadas fueran atendidas mediante una jerarquía de empelados con ciertos roles. Se implementó el patrón *Chain of responsibility* para configurar de forma más ordenada la jerarquía de los roles que debían atender las llamadas. También Se implementaron colas de información para registrar cada una de las llamadas entrantes. En total se implementaron 4 colas para las llamadas (Entrantes, pendientes, en curso y atendidas) y 1 por cada uno de los roles existentes. Cada vez que entraba una llamada se extraía una llamada de las colas de llamadas pendientes y entrantes, y se los asignaba a unos de los empleados que se almacenaban en las colas correspondientes.

Cada llamada se despachaba de manera asíncrona (haciendo uso de la clase CompletableFuture), por lo que recibíamos una respuesta inmediata y luego volvíamos a recibir una respuesta luego de 5 o 10 segundos (tiempo aleatorio).

El proyecto implementa dos clases con casos de pruebas. Uno con los casos básicos y otro verificando los puntos extras.

A continuación, especifico unas notas sobre los puntos extras:

* **Dar alguna solución sobre qué pasa con una llamada cuando no hay ningún empleado libre:** Para esto se implementó una cola de espera donde se almacenaban las llamadas si están no pudieron ser atendidas ni por un operador, supervisor o director. Posteriormente cada vez que se culminaba una llamada se revisaba la cola de llamadas de pendientes y se procesaba, luego se hacía lo mismo con la cola de llamadas entrantes solo si la anterior cola ya se había desocupado.
* **Dar alguna solución sobre qué pasa con una llamada cuando entran más de 10 llamadas concurrentes**: Para esto se implementó dos soluciones. Una fue apoyada en la cola de espera donde una llamada puede quedar esperando hasta que alguno de los 10 elementos de la cola de “En curso” se despaché y este pueda ser atendida. La segunda solución fue que, si aun así se supera la capacidad del tamaño de las colas de entrada, de espera y de en curso, la llamada se rechaza, colocándole un estado que lo identifique con dicho estado y retornándola si agregarla a ninguna cola.
* **Agregar los tests unitarios que se crean convenientes**: Se agregaron dos clases con casos de pruebas. Uno básico y otro probando los puntos extras.
* **Agregar documentación de código**: Se anexa documentación del código (javadoc), un diagrama de clases como apoyo a la documentación.