

**Infraestructura computacional**

**Ingeniería de Sistemas y Computación**

**ISIS-2203**

Daniela Mariño - 201412576

Joan Torres - 201315711

María Arévalo - 201415326

**Caso 1: Logística y Seguridad Aeroportuaria S.A**

1. **Diseño y funcionamiento**

Para el caso de estudio “Logística y Seguridad Aeroportuaria S.A” en el que se requería manejar consultas de varios clientes sobre un servidor se tomó la decisión de crear 5 clases: Admin, Buffer, Cliente, Mensaje y Servidor. A continuación se muestra como estas clases interactúan entre sí.

**1.1 Interacción Admin-Buffer**

La clase Admin contiene el main del programa. Es la encargada de procesar el archivo properties con el número de clientes, el número de servidores y el tamaño del buffer. Con estos datos, tiene que inicializar el buffer quien, a su vez, inicializa los clientes y servidores y los pone a correr.

**1.2 Interacción Buffer-Cliente**

El buffer es el encargado de almacenar temporalmente los mensajes de los clientes. Para esto, tiene una lista de mensajes a responder donde almacena todos los mensajes que necesitan respuesta. Mientras la cantidad de mensajes no sobrepase su capacidad total, este acepta mensajes y los agrega a su lista (este método es sincronizado para evitar que los mensajes se sobrescriban en una misma posición por haber entrado al mismo tiempo). Cuando todos los mensajes de un cliente tienen respuesta, el buffer saca al cliente de su cola de clientes sin terminar.

La clase Cliente tiene un número de mensajes a enviar, cuando empieza a correr, el cliente crea un mensaje e intenta enviarlo al buffer. Este agrega el mensaje a su lista si puede recibirlo, si no puede recibir el mensaje, el cliente se queda en espera activa, y cede el procesador (esto se hace mediante el método yield). Cuando el buffer haya recibido el mensaje, el cliente se duerme esperando a que el mensaje sea respondido. Cuando un mensaje es respondido, el cliente suma uno a su lista de mensajes con respuesta e intenta enviar otro mensaje. Este proceso continua hasta que el cliente haya enviado la totalidad de sus mensajes.

**1.3 Interacción Buffer-Servidor**

El servidor es el encargado de retirar los mensajes de buffer y responderlos. Mientras que el buffer aun tenga clientes y el servidor no este respondiendo ningún mensaje, el servidor intentará conseguir el primer mensaje de la lista de mensajes del buffer para responderlo (este método esta sincronizado para evitar que dos servidores intenten responder el mismo mensaje al tiempo). Si el servidor no puede retirar un mensaje del buffer, este se queda en espera activa (cediendo el procesador mediante el método yield). Después de obtener el mensaje, el servidor envía su respuesta en el mensaje que está respondiendo y deja en null el mensaje actual para intentar conseguir otro mensaje.

* 1. **Interacción Cliente-Mensaje**

Un cliente crea un mensaje, el cual guarda la información de su remitente, cuando un mensaje es respondido este le avisa a su remitente que puede despertar y continuar con sus operaciones.

* 1. **Interacción Servidor-Mensaje**

Los servidores son los encargados de responder mensajes, guardan la respuesta dentro del mensaje y dejan que este le avise al cliente.

**1.6 Interacción Buffer-Mensaje**

El buffer posee una lista de mensajes a responder, cuando un mensaje es respondido el buffer lo borra de la lista permanentemente.