**BROKER DE MENSAJES STOMP CON WEBSOCKETS + HTML5 CANVAS.**

**AUTORES:**

YORKS GOMEZ – CESAR VÁSQUEZ

# PROFESOR:

JAVIER IVAN TOQUICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

ARQUITECTURAS DE SOTFWARE - GRUPO # 1

INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2023

# INTRODUCCIÓN

# DESARROLLO

# PARTE I

# Hacemos que la aplicación HTML5/JS al ingresarle en los campos de X y Y, además de graficarlos, los publicamos en el tópico: /topic/newpoint.

# Dentro del módulo JavaScript modificamos la función de conexión/suscripción al WebSocket, para que la aplicación se suscriba al tópico "/topic/newpoint". Asociamos como 'callback' de este suscriptor una función que muestre en un mensaje de alerta el evento recibido. Como se sabe que en el tópico indicado se publicarán sólo puntos, extraemos el contenido enviado con el evento (objeto JavaScript en versión de texto), convirtiéndolo en objeto JSON, y extraemos de éste sus propiedades (coordenadas X y Y).

# 

# Una captura de pantalla de un celular Descripción generada automáticamente

Compilamos y ejecutamos nuestra aplicación. Abrimos la aplicación en varias pestañas diferentes. Ingresamos los datos, damos click a la acción del botón, y verificamos que en todas la pestañas se haya lanzado la alerta con los datos ingresados.

# Texto Descripción generada automáticamente

# Interfaz de usuario gráfica, Texto, Aplicación, Sitio web Descripción generada automáticamente

# PARTE II

# Hacemos que el 'callback' asociado al tópico /topic/newpoint en lugar de mostrar una alerta, dibuje un punto en el canvas en las coordenadas enviadas con los eventos recibidos. Dibujamos un círculo de radio 1.

# 

# Texto Descripción generada automáticamente

# Texto Descripción generada automáticamente

# Ejecutamos la aplicación en varios navegadores (y si puede en varios computadores, accediendo a la aplicación mediante la IP donde corre el servidor). Compruebe que a medida que se dibuja un punto, el mismo es replicado en todas las instancias abiertas de la aplicación.

# Forma Descripción generada automáticamente

# Interfaz de usuario gráfica Descripción generada automáticamente con confianza media

# Efectivamente se dibuja de manera correcta en los diferentes navegadores.

# PARTE III

# Agregamos un campo en la vista, en el cual el usuario pueda ingresar un número. El número corresponderá al identificador del dibujo que se creará.

# 

# Modificamos la aplicación para que, en lugar de conectarse y suscribirse automáticamente, lo haga a través de botón 'conectarse'. Éste, al oprimirse debe realizar la conexión y suscribir al cliente a un tópico que tenga un nombre dinámico, asociado el identificador ingresado, por ejemplo: /topic/newpoint.25, topic/newpoint.80, para los dibujos 25 y 80 respectivamente.

# Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto Descripción generada automáticamente

# De la misma manera, hacemos que las publicaciones se realicen al tópico asociado al identificador ingresado por el usuario.

# Rectifique que se puedan realizar dos dibujos de forma independiente, cada uno de éstos entre dos o más clientes.

# DIAGRAMA DE COMPONENTES

# CONCLUSIONES

# Comunicación en tiempo real: La combinación de websockets y STOMP permite una comunicación bidireccional en tiempo real entre el cliente y el servidor. Esto es especialmente útil en aplicaciones web en las que se requiere una comunicación en tiempo real, como en juegos en línea, sistemas de colaboración, etc.

# Visualización de datos: La capacidad de enviar datos en tiempo real desde el servidor al cliente y visualizarlos en un lienzo de HTML5 Canvas puede ser útil en aplicaciones que necesitan mostrar datos en tiempo real de una manera más visual, como en monitoreo de datos en tiempo real, gráficos y tableros de control.

# Integración de sistemas: STOMP es un protocolo de mensajería interoperable que puede ser utilizado por diferentes sistemas. Por lo tanto, el uso de un broker de mensajes STOMP con websockets puede permitir la integración de diferentes sistemas que necesitan comunicarse entre sí en tiempo real.

# Escalabilidad: Un broker de mensajes STOMP con websockets puede ser una solución escalable para aplicaciones que necesitan manejar grandes cantidades de mensajes en tiempo real. Al utilizar un broker de mensajes, se puede delegar la tarea de distribuir los mensajes a través de múltiples servidores, lo que puede mejorar la capacidad de procesamiento y reducir la carga en un solo servidor.