

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**2CM3**

**PROFESOR: TECLA PARRA ROBERTO**

**PRÁCTICA 6 SOCKETS SERVIDORES (CHATBOT)**

**VÁZQUEZ MORENO MARCOS OSWALDO 2016601777**

**FECHA DE ENTREGA: 19 DE MARZO DE 2018**

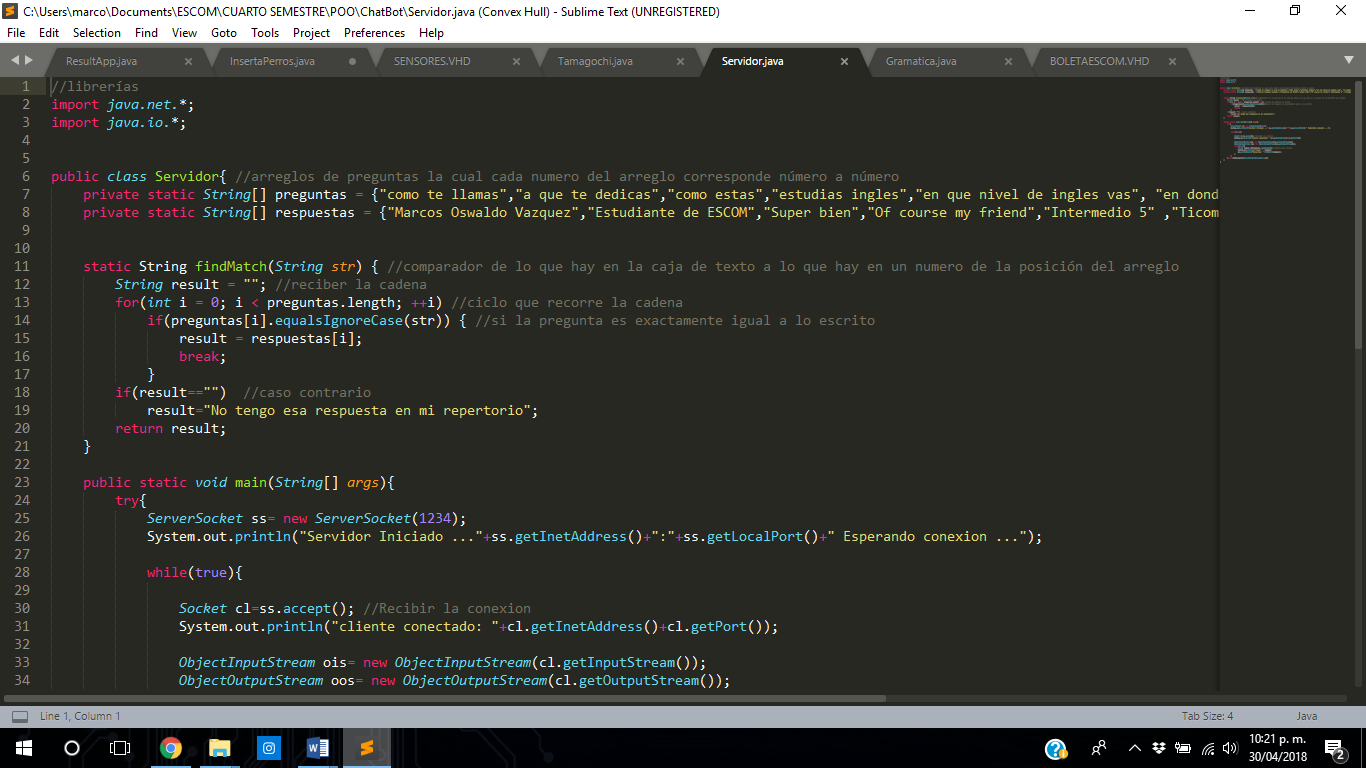
**OBJETIVO**

Realizar un programa en lenguaje de programación Java codificando un cliente y un servidor que interactúen del siguiente modo el cliente envía una pregunta al servidor y el servidor envía una respuesta al cliente. El servidor puede almacenar al menos 10 preguntas y 10 respuestas predefinidas (se pueden usar 2 arreglos o un HashMap).

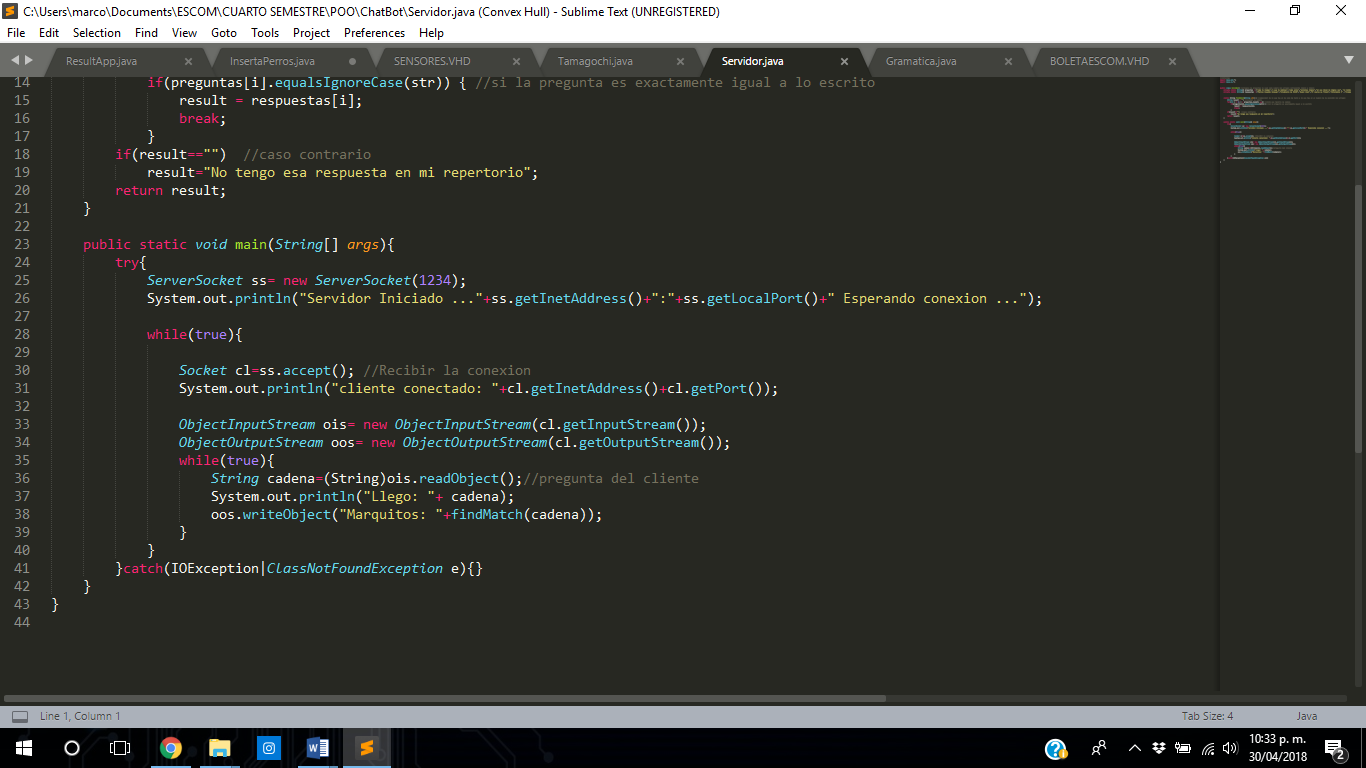
**DESARROLLO**

Primeramente, agregamos todas las librerías y extensiones de Java para poder llevar acabo nuestra conexión de Sockets Servidores.

Posteriormente creamos una clase principal llamada *Servidor* en donde declaramos dos arreglos de tipo cadena con valor dinámico, uno para preguntas y otro para respuestas, cada posición de la pregunta corresponde a su respuesta y viceversa, después tenemos a nuestra función *findMatch*  también de tipo cadena en donde va a comparar si la pregunta ingresada por el usuario es exactamente igual a una de las que se tiene en el arreglo para poder lanzar la respuesta correspondiente, en caso contrario muestra un mensaje diciendo “No tengo esa respuesta en mi repertorio”.



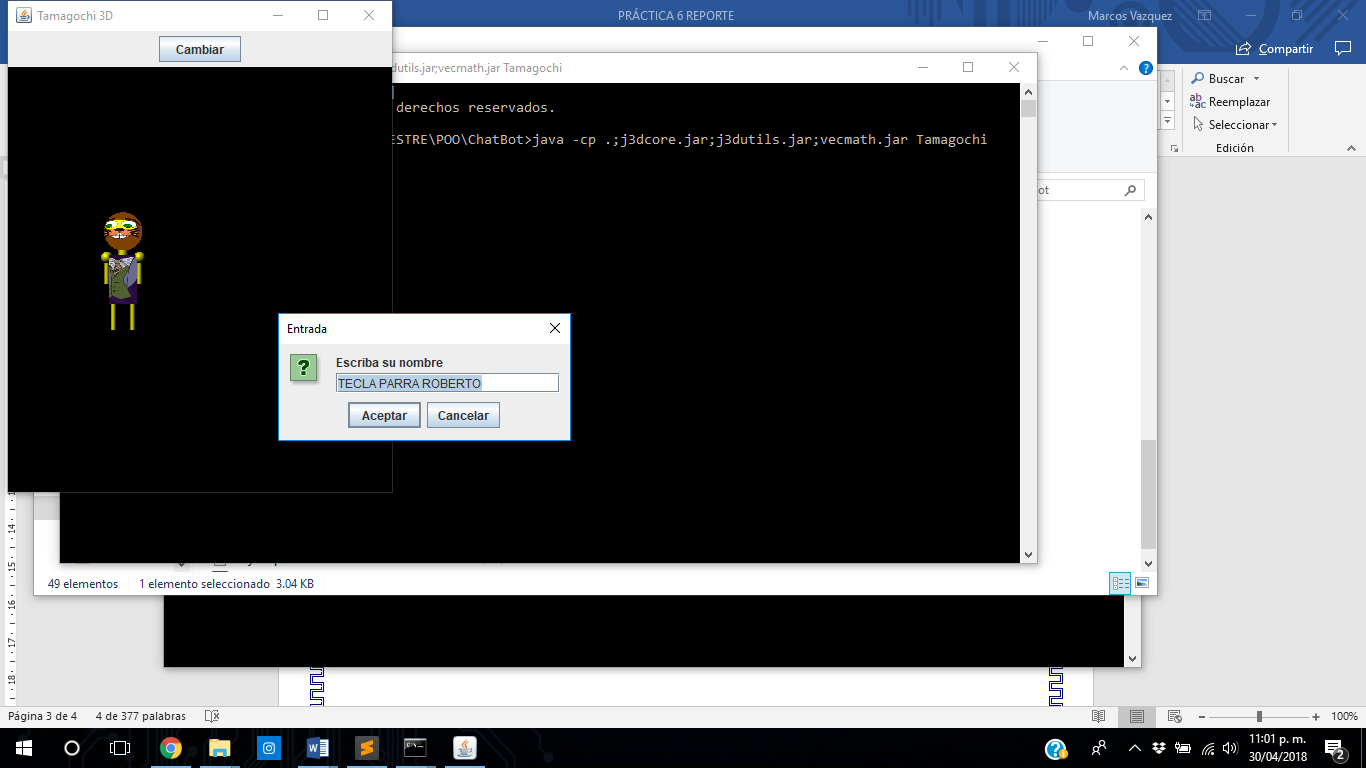
A continuación, nuestro método principal en el cual el server busca la conexión y la establece en caso de encontrar al cliente mediante el método OutputStream y en caso de recibir la pregunta escribe en consola que recibió pregunta, todo esto mientras la pregunta se encuentre dentro del arreglo de preguntas, en la línea número 38 asignas el nombre de la persona que es el cliente en este caso. Por último, se tiene la excepción.

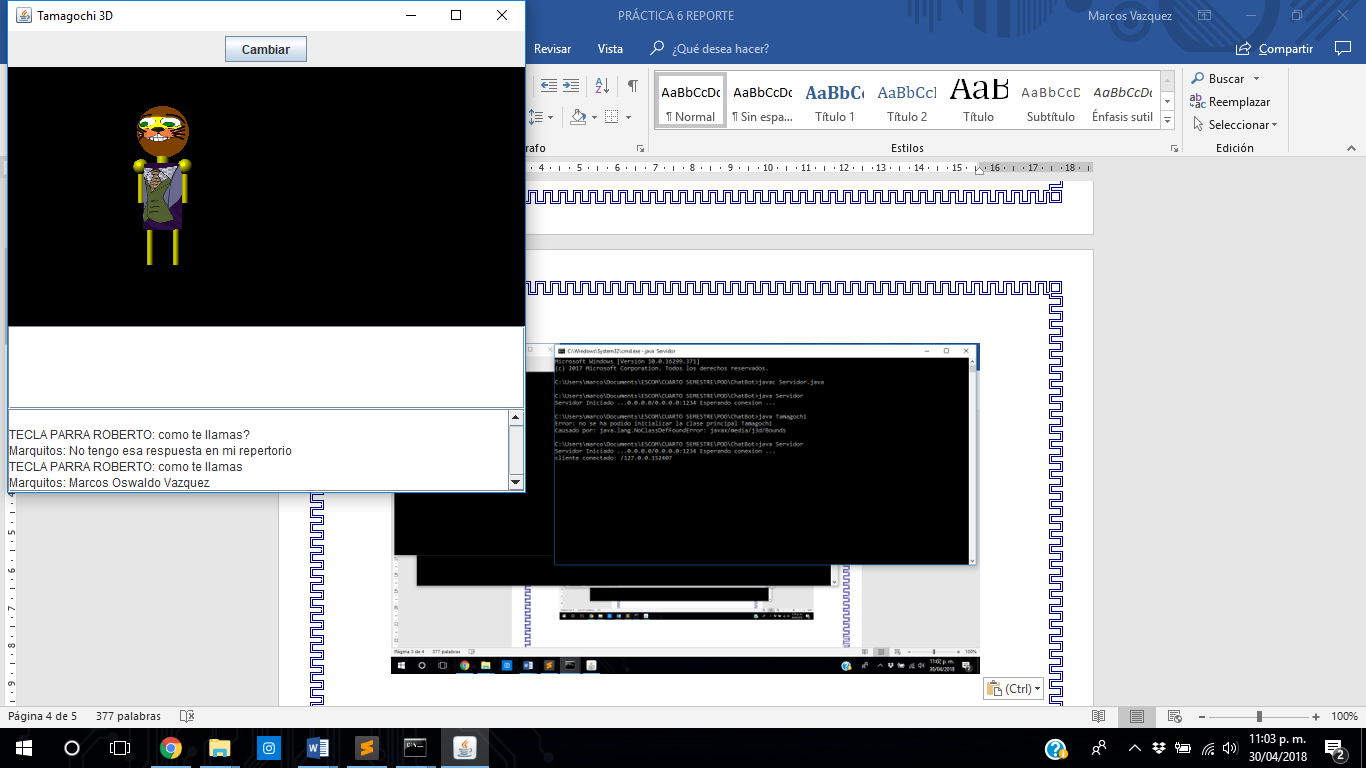
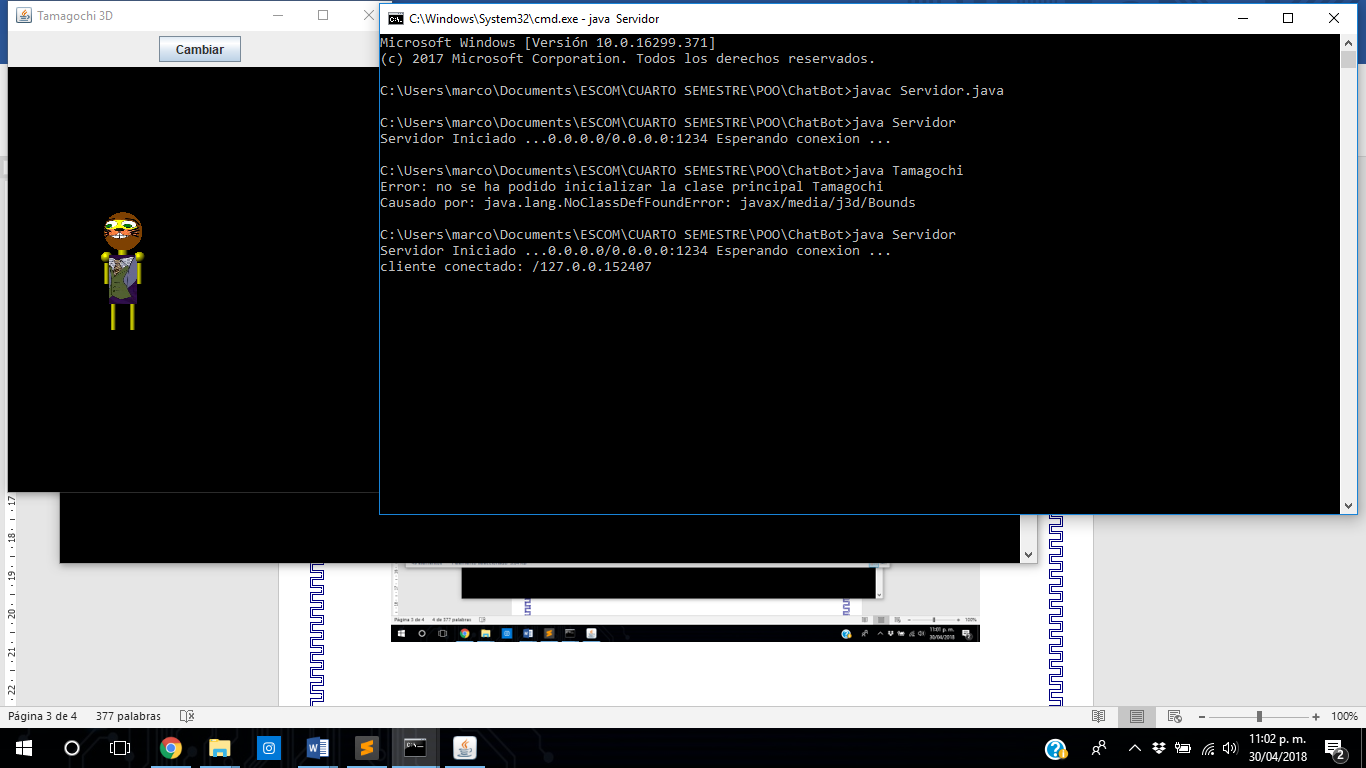


Cabe destacar que todo este código anterior se ejecuta con el Tamagochi.java de la práctica anterior pues solo es usado para el entorno gráfico.

**CONCLUSIÓN**

En conclusión, puedo decir que fue una práctica bastante creativa, con buen aprendizaje hacia nosotros ya que existió un juego en el que le hacías preguntas a un tamagochi y el te respondía siempre y cuando la respuesta esté en el arreglo, así como la pregunta, creando el motivo perfecto para tener un adecuado camino creado entre el cliente y el servidor.





Para obtener los resultados esperados ejecutamos estas sentencias:

java -cp .;j3dcore.jar;j3dutils.jar;vecmath.jar Tamagochi

java Servidor